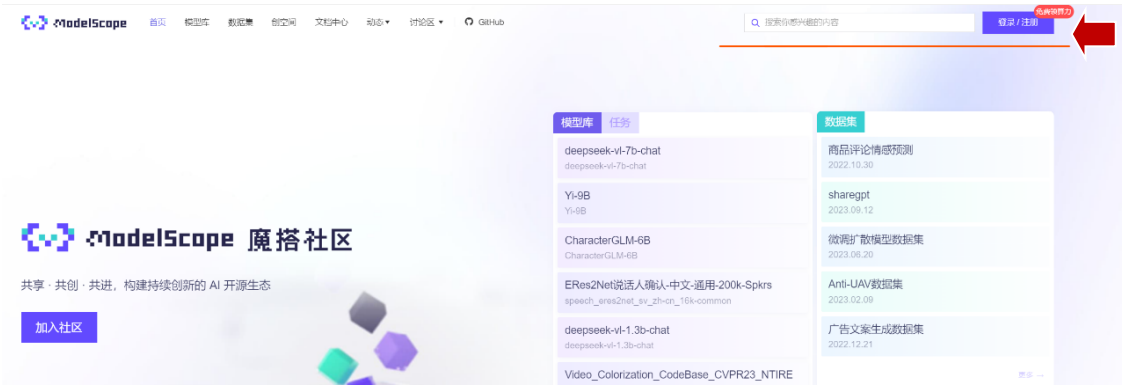
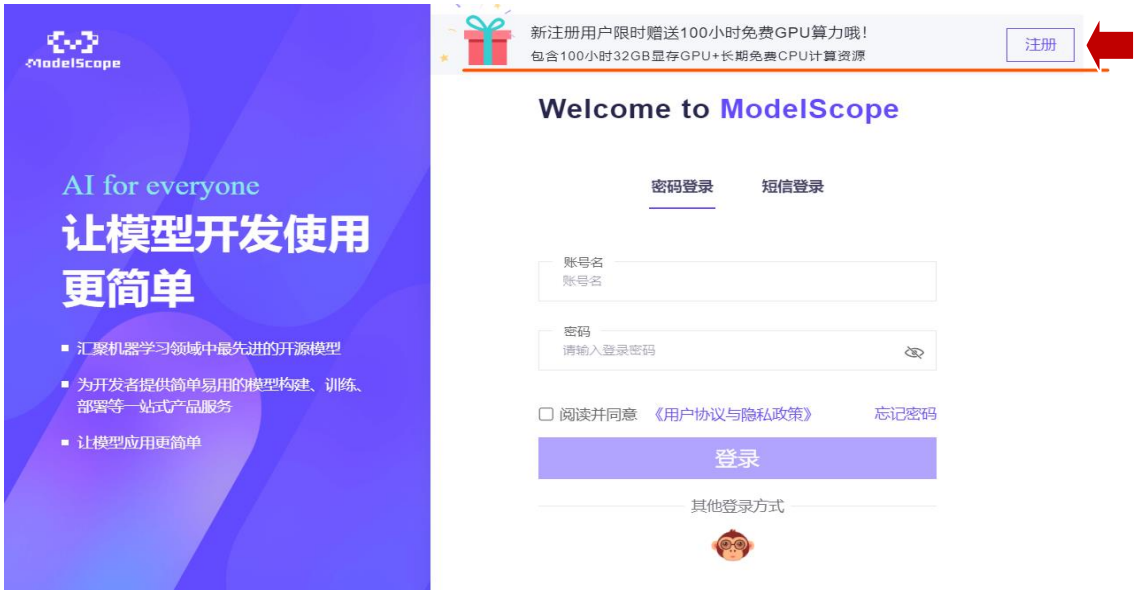
动手实验

一、 注册&登录 ModelScope

1. 进入 ModelScope（https://www.modelscope.cn/home），右上角点击完成新用户的注册；



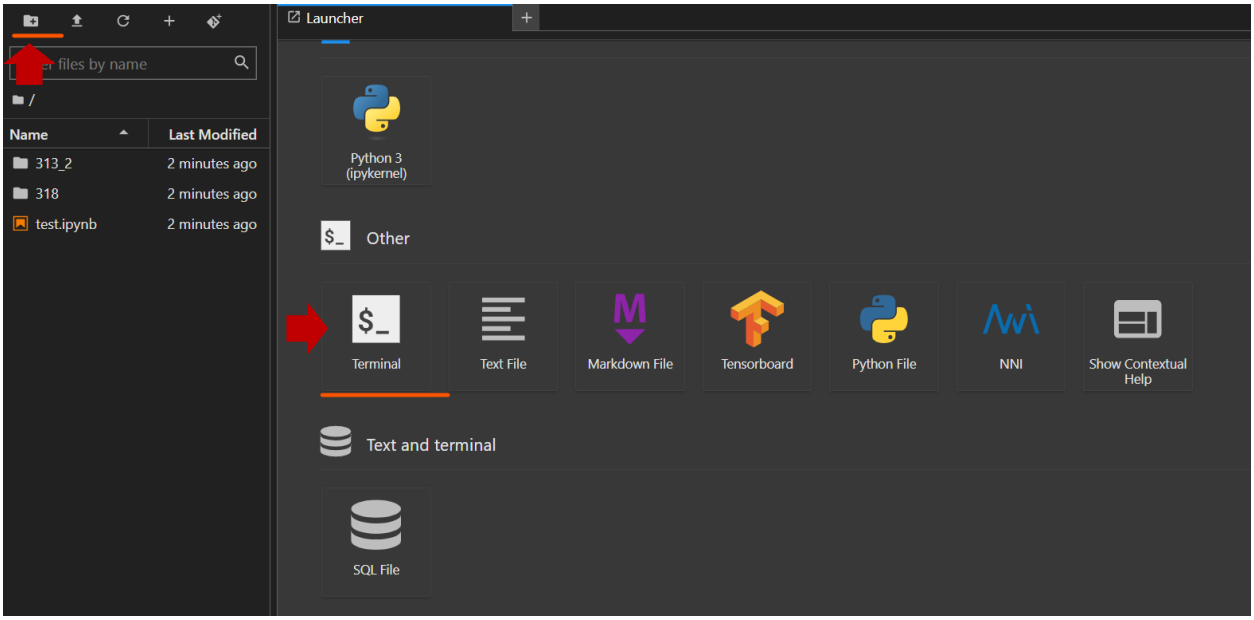


2. 注册完成后，登录 modelscope，进入首页，请确定已经绑定阿里云账号，并已获得免费的配套云计算资源。启动 CPU 服务器：

3. 启动成功后，点击“查看”

二、环境配置

1. 打开 terminal



2. 环境搭建:

环境目录/opt/conda/envs 下面新建文件夹 itrex:

```
cd  /opt/conda/envs
mkdir itrex
```

拷贝镜像文件至 itrex 目录(任选一个下载即可):

```
wget https://idz-ai.oss-cn-hangzhou.aliyuncs.com/LLM/itrex.tar.gz

wget https://filerepo.idzcn.com/LLM/itrex.tar.gz
```

解压文件:

```
tar -zxvf itrex.tar.gz -C itrex/
```

激活环境:

```
conda activate itrex
```

安装对应的 kernel:

```
python -m ipykernel install --name itrex
```

三、下载示例代码

切换至工作目录:

```
cd /mnt/workspace
```

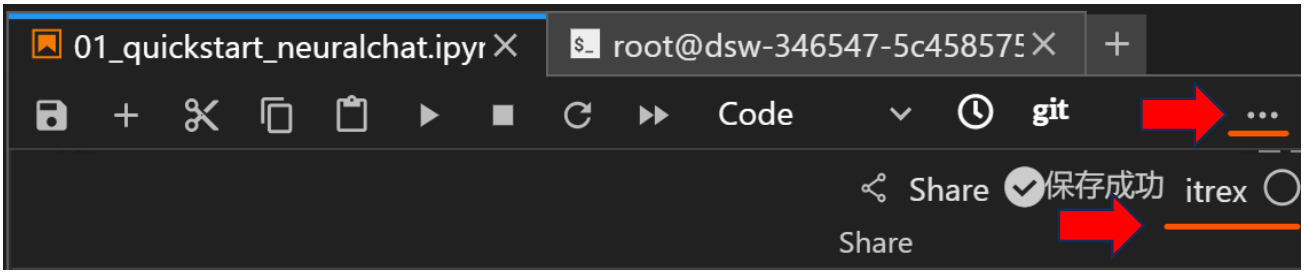
下载 notebook 代码文件(任选一个下载即可)

```
wget https://idz-ai.oss-cn-
hangzhou.aliyuncs.com/LLM/01_quickstart_neuralchat.ipynb

wget https://filerepo.idzcn.com/LLM/01_quickstart_neuralchat.ipynb

wget https://raw.githubusercontent.com/intel/intel-extension-for-
transformers/main/intel_extension_for_transformers/neural_chat/docs/n
otebooks/workshop/01_quickstart_neuralchat.ipynb
```

双击.ipynb 文件，选择 itrex 作为 kernel：



四、运行示例代码

Customizing your Chatbot

Plugin: Retrieval

Without the retrieval plugin, the output of the chatbot gives the wrong answer.

```
from intel_extension_for_transformers.neural_chat import build_chatbot, PipelineConfig
config = PipelineConfig(model_name_or_path='Intel/neural-chat-7b-v3-1')
chatbot = build_chatbot(config)
response = chatbot.predict(query="Who won Super Bowl 58 and what was the score?")
print(response)
```

1. Customizing your Chatbot :Without RAG：

    1) 下载中文大模型至本地：

```
git clone https://www.modelscope.cn/ZhipuAI/chatglm2-6b.git
```

    2) 将 model_name_or_path 修改为本地路径，修改代码如下

```
# Build chatbot with INT4 weight-only quantization, computations in
AMX INT8
from intel_extension_for_transformers.neural_chat import
build_chatbot, PipelineConfig
```

```
from intel_extension_for_transformers.transformers import
WeightOnlyQuantConfig
from intel_extension_for_transformers.neural_chat.config import
LoadingModelConfig

config = PipelineConfig(model_name_or_path="./chatglm2-6b",
optimization_config=WeightOnlyQuantConfig(compute_dtype="int8",
weight_dtype="int4_fullrange"),
loading_config=LoadingModelConfig(use_neural_speed=False))

chatbot = build_chatbot(config)

# Perform inference/generate a response
response = chatbot.predict(query="cnvrg.io 网站是由谁创建的？")
print(response)
```

3) 点击左侧按钮运行;

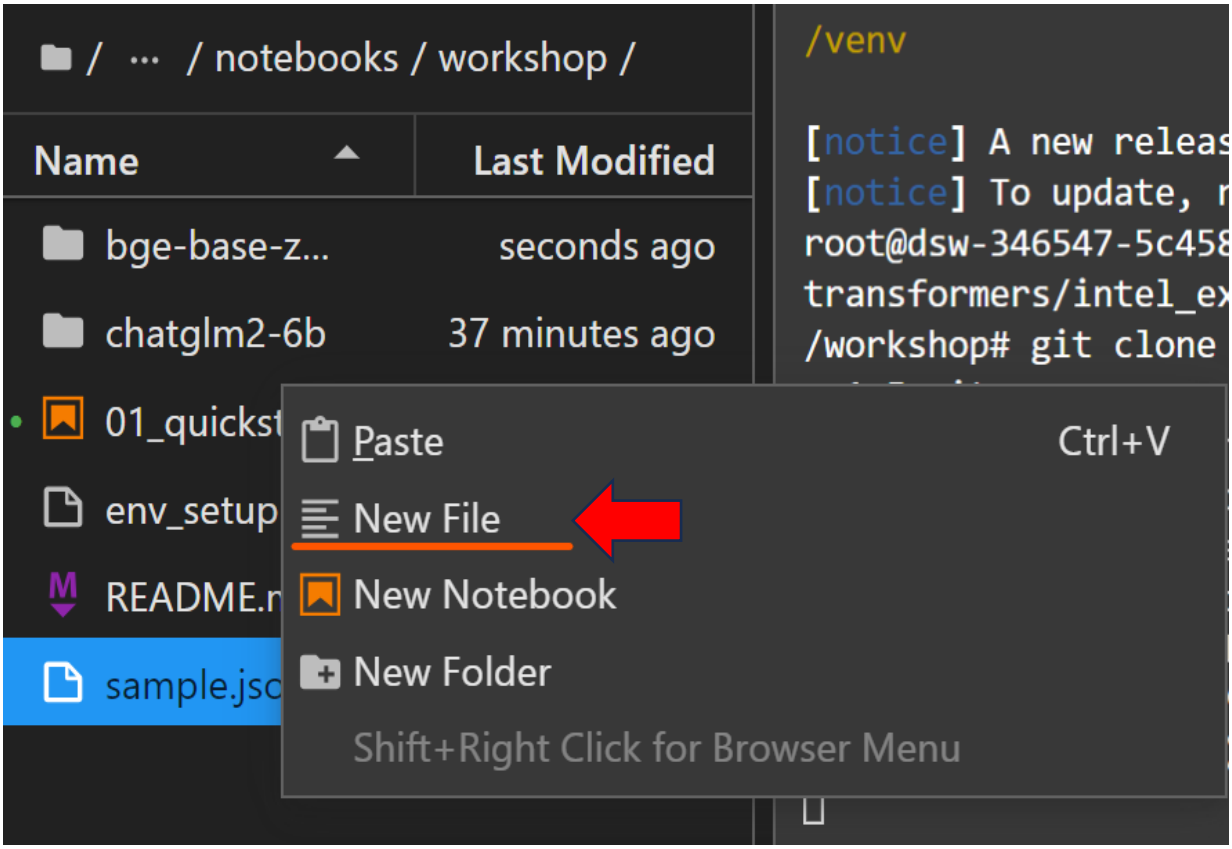
2. Customizing your Chatbot： Plugin: Retrieval

1) 下载 embedding 模型至本地:

```
git clone https://www.modelscope.cn/AI-ModelScope/bge-base-zh-
v1.5.git
```

2) 准备 input 文件 sample.jsonl

i. 新建文件 sample.jsonl：右键->New File



ii. 双击文件，在文件里面填入如下内容，保存文件。

```
{"content": "cnvrg.io 网站由 Yochay Ettun 和 Leah Forkosh Kolben 创
建.", "link": 0}
```

3) 修改代码如下：

```
import time
# Build chatbot with retrieval
from intel_extension_for_transformers.neural_chat import
PipelineConfig
from intel_extension_for_transformers.neural_chat import
build_chatbot
from intel_extension_for_transformers.neural_chat import plugins
```

```
from intel_extension_for_transformers.transformers import
WeightOnlyQuantConfig
from intel_extension_for_transformers.neural_chat.config import
LoadingModelConfig

plugins.retrieval.enable=True
plugins.retrieval.args['embedding_model'] = "./bge-base-zh-v1.5"
plugins.retrieval.args["input_path"]="./sample.jsonl"
config = PipelineConfig(model_name_or_path='./chatglm2-6b',
plugins=plugins,
optimization_config=WeightOnlyQuantConfig(compute_dtype="int8",
weight_dtype="int4_fullrange"),
loading_config=LoadingModelConfig(use_neural_speed=False))

chatbot = build_chatbot(config)

st = time.time()
response = chatbot.predict(query="cnvrg.io 网站是由谁创建的？")
et = time.time()
print("predict 时间：{}s".format(et-st))

print(response)

plugins.retrieval.enable=False # disable retrieval
```

五、常见问题

1. No module named gguf
   解决方法：pip install gguf
2. GENETIC ERROR， retriever build 失败
   解决方法：保存文件、restart kernel、关闭文件、重新打开、restart kernel、等 2-3 分钟

3. modelscope 下载失败：
   解决方法：等 2-3 分钟，在终端里面下载；
4. Finetune 时显示数据集不合法：
   解决方法：修改 cnvrg_dataset 里面的文件内容后等 2-3 分钟再次运行；