How to train and (fine-tune) an LLM on VSC5, (short demo)

Soner Steiner

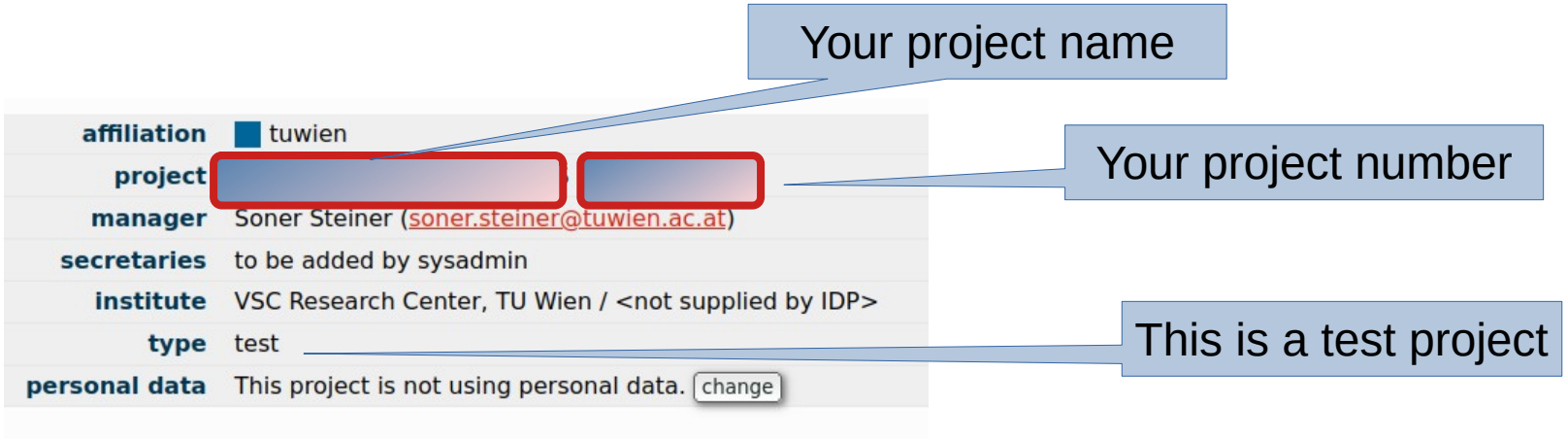
VSC Research Center / EuroCC Austria, TU-WIEN, Vienna, Austria
soner dot steiner at tuwien dot ac dot at

Introduction

- Training your LLM model on VSC5
- Your project at VSC5
- Interactive sessions
- Slurm scripts
- Fine tune a (BERT) model
- LLM_webinar_EuroCC_github

Training your model on VSC5

- Talk to us → get a project at the cluster, managed by euroCC or VSC stuff



Training your model on VSC5

- For login, you need a TU-Wien, Uni-Wien IP
- Either VPN or we need to give you access from a specific IP-adress

```
link/none
inet 128.131.228.155/32 scope global tun0
```

tuwien ip number

for that we need a telnumber

```
:>$ ssh mithridates@vsc5.vsc.ac.at
(mithridates@vsc5.vsc.ac.at) Password:
(mithridates@vsc5.vsc.ac.at) sms challenge sent, please enter OTP:
Last login: Wed Oct  4 16:26:08 2023 from 128.131.228.155
```

Training your model on VSC5

- Welcome to the VSC5

```
WELCOME TO VSC-5
================================================================

To see available partitions:    sinfo
To submit jobs type:            sbatch job_script
To view the job status type:    squeue
Slurm documentation:            http://slurm.schedmd.com/
```

```
partition                                   QOS
-----------------------------------------------
cascadelake_0384                        cascadelake_0384
zen2_0256_a40x2                         zen2_0256_a40x2
zen3_0512_a100x2                        zen3_0512_a100x2
zen3_0512                               zen3_0512,zen3_0512_devel
zen3_1024                               zen3_1024
zen3_2048                               zen3_2048
```

We want to use GPUs

Training your model on VSC5

- **An interactive session with the GPU**

allocate a A100 GPU
for an interactive session

```
zen mithridates@l53:~/llm_models$ salloc -N 1 -p zen3_0512_a100x2 --qos zen3_0512_a100x2 --gres=gpu:2
salloc: Pending job allocation 1204101
salloc: job 1204101 queued and waiting for resources
salloc: job 1204101 has been allocated resources
salloc: Granted job allocation 1204101
salloc: Waiting for resource configuration
salloc: Nodes n3073-006 are ready for job
zen mithridates@l53:~/llm_models$ ssh -X mithridates@n3073-006
Warning: Permanently added 'n3073-006,10.191.73.6' (ECDSA) to the list of known hosts.
mithridates@n3073-006's password:
cuda-zen mithridates@n3073-006:~$
```

Cuda spack env

login to the GPU node

Training your model on VSC5

- **An interactive session with the GPU**

Source the cuda env variables

```
cuda-zen mithridates@l52:~$ source /opt/sw/cuda-zen/spack-0.19.0/share/spack/setup-env.sh
cuda-zen mithridates@l52:~$ module load cuda/11.8.0-gcc-12.2.0-knnuyxt
```

load the required modules

Training your model on VSC5

- An interactive session with the GPU

```
cuda-zen mithridates@n3073-006:~/llm_models$ nvidia-smi
Tue Oct  3 10:15:17 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 510.39.01    Driver Version: 510.39.01    CUDA Version: 11.6     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A100-PCI...  Off  | 00000000:01:00.0 Off |                  Off |
| N/A   40C    P0    38W / 250W |      0MiB / 40960MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A100-PCI...  Off  | 00000000:81:00.0 Off |                  Off |
| N/A   35C    P0    39W / 250W |      0MiB / 40960MiB |     38%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

check the GPU you are on

Training your model on VSC5

- A simple model for, which I need the resources on VSC5
- Splitted the work into four steps

- **Prepare the data**
- **Do the training**
- **Test the model**
- **Save the model**

Training your model on VSC5

- There are several ways to do the work
- I will show the interactive one first

prepare the data

```
import os
import tensorflow as tf

filepath="shakespeare.txt"
with open(filepath) as f:
    shakes_text = f.read()

print(shakes_text[100:180])
"".join(sorted(set(shakes_text.lower())))

text_vec_layer = tf.keras.layers.TextVectorization(split="character",
                                                   standardize="lower")
text_vec_layer.adapt([shakes_text])
encoded = text_vec_layer([shakes_text])[0]

encoded -= 2
n_tokens = text_vec_layer.vocabulary_size() - 2
dataset_size = len(encoded)

def to_dataset(sequence, length, shuffle=False, seed=None, batch_size=32):
    ds = tf.data.Dataset.from_tensor_slices(sequence)
    ds = ds.window(length + 1, shift=1, drop_remainder=True)
    ds = ds.flat_map(lambda window_ds: window_ds.batch(length + 1))
```

Training your model on VSC5

prepare the data

```
:>$ python3.9
Python 3.9.7 (default, Apr 21 2022, 08:39:11)
[GCC 8.5.0 20210514 (Red Hat 8.5.0-10)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> exec(open("shakes_prepare.dat.py").read() )
```

Training your model on VSC5

create and train your model

```
 8 def create_model():
 9         model = tf.keras.Sequential([
10         tf.keras.layers.Embedding(input_dim=n_tokens, output_dim=16),
11          tf.keras.layers.GRU(128, return_sequences=True),
12         tf.keras.layers.Dense(n_tokens, activation="softmax")
13         ])
14
15         model.compile(loss="sparse_categorical_crossentropy", optimizer="nadam",
16                       metrics=["accuracy"])
17
18         return model
19
20 shakes_model = create_model()
21 shakes_model.summary()
22
23 checkpoint_path = "shakes_model/cp.ckpt"
24 checkpoint_dir = os.path.dirname(checkpoint_path)
25
26 # Create a callback that saves the model's weights
27 # that is very important
28 cp_callback = tf.keras.callbacks.ModelCheckpoint(
29               filepath=checkpoint_path,
30               monitor="val_accuracy",
31               save_best_only=True,
32               save_weights_only=True,
33               verbose=1)
34
35 # Train the model with the new callback
36 shakes_model.fit(train_set,
```

Training your model on VSC5

test your model

```
43 def next_char(text, temperature=1):
44     y_proba = shakes_model.predict([text])[0, -1:]
45     rescaled_logits = tf.math.log(y_proba) / temperature
46     char_id = tf.random.categorical(rescaled_logits, num_samples=1)[0, 0]
47     return text_vec_layer.get_vocabulary()[char_id + 2]
48
49 def next_char_ss(text, my_model, temperature=1):
50     y_proba = my_model.predict([text])[0, -1:]
51     rescaled_logits = tf.math.log(y_proba) / temperature
52     char_id = tf.random.categorical(rescaled_logits, num_samples=1)[0, 0]
53     return text_vec_layer.get_vocabulary()[char_id + 2]
54
55 def extend_text(text, my_model, n_chars=50, temperature=1):
56     for _ in range(n_chars):
57         text += next_char_ss(text, my_model, temperature)
58     return text
59
60 print(extend_text("To be or not to be", temperature=0.5))
61
```

Training your model on VSC5

test your model

to be or not to be so in the world and the strangeness
to see the wo

one can get better results by augmenting the train data set

to be or not to be that is the …..

Training your model on VSC5

reload your model after training

```
20 def create_model():
21     model = tf.keras.Sequential([
22     tf.keras.layers.Embedding(input_dim=n_tokens, output_dim=16),
23     tf.keras.layers.GRU(128, return_sequences=True),
24     tf.keras.layers.Dense(n_tokens, activation="softmax")
25     ])
26     model.compile(loss="sparse_categorical_crossentropy", optimizer="nadam",metrics=["accuracy"
27     return model
28
29 shakes_model = create_model()
30 shakes_model.summary()
31
32 checkpoint_path = "shakes_model/cp.ckpt"
33 checkpoint_dir = os.path.dirname(checkpoint_path)
34
35 # Loads the weights
36 shakes_model.load_weights(checkpoint_path)
37
```

Training your model on VSC5

of course one can also use slurm scripts

```
#!/bin/sh
#SBATCH -J jobname
#SBATCH -N 1
#SBATCH --partition=zen3_0512_a100x2
#SBATCH --qos zen3_0512_a100x2
#SBATCH --gres=gpu:2

cuz
source /opt/sw/cuda-zen/spack-0.19.0/share/spack/setup-env.sh
module load cuda/11.8.0-gcc-12.2.0-knnuyxt

python3.9 train_model.py


```

Training your model on VSC5

of course one can also use slurm scripts

```
:>$ sbatch train_model_script
Submitted batch job 1336279
:>$
:>$ squeue -u mithridates
             JOBID             PARTITION     NAME     USER ST       TIME  NODES     NODELIST(REASON)
           1336279     zen3_0512_a100x2  jobname mithrida PD       0:00      1         (QOSGrpGRES)
:>$
```

Training your model on VSC5

a few words on ML, DL, AI and the frameworks in general

- Take your time for data preparation
- Test before you submit a big job → interactive sessions
- Don’t mix the frameworks, use just one, e.g.: tensorflow, pytorch, scikit, ...
- Have big datasets more then 100 GB → get in touch with us
- Multi-GPU Training → get in touch with us
- At the moment everything is changing very fast, e.g. nvidia has its own versions, intel has extensions of scikit-learn, ...

Small live demo at our cluster

```
WELCOME TO VSC-5
================================================================================

To see available partitions:     sinfo
To submit jobs type:             sbatch job_script
To view the job status type:     squeue
Slurm documentation:             http://slurm.schedmd.com/
```

Fine tune a (Bert) model

- There are several sources for pretrained models, e.g.:
- Intel’s modelzoo: modelzoo
- tensorflow-hub
- kaggle
- modelzoo

Fine tune a (Bert) model

- Choose your model
- Data preparation
- Check the model architecture
- Hyperparameter tuning
- Evaluation

Fine tune a (Bert) model

- Fine tuning a pre-trained model can be very powerful
- We can use the same model architecture to new datasets
- The dataset should be similar to the original dataset the pretrained model was trained on

Summary Outlook

- Contact us, if you want to use our cluster
- Ask for a test-project
- Have big datasets more then 100 GB → get in touch with us
- Multi-GPU Training → get in touch with us

References

- LLM_webinar_EuroCC_githubLLM_webinar_EuroCC_github
  https://github.com/sonersteiner/20231009_LLM_webinar_EuroCC
- ageron-ml3-github
  https://github.com/ageron/handson-ml3
- dl-python-chollet-github
  https://github.com/fchollet/deep-learning-with-python-notebooks