AAI520_Final_Group_AutoTrain_LLM

October 16, 2023

0.1 AAI-520

0.2 Final Project - Group 6

0.3 Chatbot for Movie Info utilizing the Cornell Movie Dialogs Corpus

This Jupyter Notebook is used for training a Meta LLaMa LLM. It is used to train the LLM using the Huggingface AutoTrain library.

This is a modified version of the Huggingface training notebook found on the Github repo: https://github.com/huggingface/autotrain-advanced

Install the libraries needed to run the HuggingFace AutoTrain utility.

[ ]:
```
#@title  AutoTrain LLM
#@markdown In order to use this colab
#@markdown - upload train.csv to a folder named `data/`
#@markdown - train.csv must contain a `text` column
#@markdown - choose a project name if you wish
#@markdown - change model if you wish, you can use most of the text-generation␣
 ↪models from Hugging Face Hub
#@markdown - add huggingface information (token and repo_id) if you wish to␣
 ↪push trained model to huggingface hub
#@markdown - update hyperparameters if you wish
#@markdown - click `Runtime > Run all` or run each cell individually

import os
!pip install -U autotrain-advanced > install_logs.txt
!autotrain setup > setup_logs.txt
```

Configuration for the HuggingFace AutoTrain. You must enter a "Write" HuggingFace token for the utility to pull and publish the model(s).

Enter a new model_name where the utility will publish the fine tuned model.

The settings will affect the training performance and time for training. We have tested with low epoch for quick iteration and have found larger epoch to perform better train but is very time consuming.

We must run the LLM on a GPU so we have moved our notebooks to Google Colab. The Pro Colab subscription has allowed us to run the V100 GPUs which have improved training time.

```
[ ]: #@markdown ---
#@markdown #### Project Config
#@markdown Note: if you are using a restricted/private model, you need to enter your Hugging Face token in the next step.
project_name = 'my_autotrain_llm' # @param {type:"string"}
model_name = 'meta-llama/Llama-2-7b-chat-hf' # @param {type:"string"}

#@markdown ---
#@markdown #### Push to Hub?
#@markdown Use these only if you want to push your trained model to a private repo in your Hugging Face Account
#@markdown If you dont use these, the model will be saved in Google Colab and you are required to download it manually.
#@markdown Please enter your Hugging Face write token. The trained model will be saved to your Hugging Face account.
#@markdown You can find your token here: https://huggingface.co/settings/tokens
push_to_hub = True # @param ["False", "True"] {type:"raw"}
hf_token = "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" #@param {type:"string"}
repo_id = "username/enter_new_repo_name_here" #@param {type:"string"}

#@markdown ---
#@markdown #### Hyperparameters
learning_rate = 2e-4 # @param {type:"number"}
num_epochs = 9 #@param {type:"number"}
batch_size = 4 # @param {type:"slider", min:1, max:32, step:1}
block_size = 1024 # @param {type:"number"}
trainer = "sft" # @param ["default", "sft"] {type:"raw"}
warmup_ratio = 0.1 # @param {type:"number"}
weight_decay = 0.01 # @param {type:"number"}
gradient_accumulation = 4 # @param {type:"number"}
use_fp16 = True # @param ["False", "True"] {type:"raw"}
use_peft = True # @param ["False", "True"] {type:"raw"}
use_int4 = True # @param ["False", "True"] {type:"raw"}
lora_r = 16 #@param {type:"number"}
lora_alpha = 32 #@param {type:"number"}
lora_dropout = 0.05 #@param {type:"number"}

os.environ["PROJECT_NAME"] = project_name
os.environ["MODEL_NAME"] = model_name
os.environ["PUSH_TO_HUB"] = str(push_to_hub)
os.environ["HF_TOKEN"] = hf_token
os.environ["REPO_ID"] = repo_id
os.environ["LEARNING_RATE"] = str(learning_rate)
os.environ["NUM_EPOCHS"] = str(num_epochs)
os.environ["BATCH_SIZE"] = str(batch_size)
os.environ["BLOCK_SIZE"] = str(block_size)
os.environ["WARMUP_RATIO"] = str(warmup_ratio)
```

```
os.environ["WEIGHT_DECAY"] = str(weight_decay)
os.environ["GRADIENT_ACCUMULATION"] = str(gradient_accumulation)
os.environ["USE_FP16"] = str(use_fp16)
os.environ["USE_PEFT"] = str(use_peft)
os.environ["USE_INT4"] = str(use_int4)
os.environ["LORA_R"] = str(lora_r)
os.environ["LORA_ALPHA"] = str(lora_alpha)
os.environ["LORA_DROPOUT"] = str(lora_dropout)
```

1 Add the huggingface write token

You must used a Write hf token

[ ]:
```
!huggingface-cli login
```

Run the HuggingFace AutoTrain utility and publish the fine tuned model to the desired repository. Note, that this is a PEFT fine tuned adapter model and will not contain the 30+ gigabyte of base model. We will attempt to create a merged model in one of the other notebooks.

[ ]:
```
!autotrain llm \
--train \
--model ${MODEL_NAME} \
--project-name ${PROJECT_NAME} \
--data-path data/ \
--text-column text \
--lr ${LEARNING_RATE} \
--batch-size ${BATCH_SIZE} \
--epochs ${NUM_EPOCHS} \
--block-size ${BLOCK_SIZE} \
--use-int4 \
--use-peft \
$( [[ "$PUSH_TO_HUB" == "True" ]] && echo "--push-to-hub --token ${HF_TOKEN}␣
↪--repo-id ${REPO_ID}" )
```

1.1 Optionally download from Google Colab

[ ]:
```
# !zip -r /content/my_autotrain_llm.zip /content/my_autotrain_llm
```

[ ]:
```
# from google.colab import files
# files.download("/content/my_autotrain_llm.zip")
```