Multiple Choice

Fine-tuning on SWAG with the Trainer

`run_swag` allows you to fine-tune any model from our hub (as long as its architecture as a `ForMultipleChoice` version in the library) on the SWAG dataset or your own csv/jsonlines files as long as they are structured the same way. To make it works on another dataset, you will need to tweak the `preprocess_function` inside the script.

```
python examples/multiple-choice/run_swag.py \
--model_name_or_path roberta-base \
--do_train \
--do_eval \
--learning_rate 5e-5 \
--num_train_epochs 3 \
--output_dir /tmp/swag_base \
--per_gpu_eval_batch_size=16 \
--per_device_train_batch_size=16 \
--overwrite_output
```

Training with the defined hyper-parameters yields the following results:

```
***** Eval results *****
eval_acc = 0.8338998300509847
eval_loss = 0.44457291918821606
```

With Accelerate

Based on the script run_swag_no_trainer.py.

Like `run_swag.py`, this script allows you to fine-tune any of the models on the hub (as long as its architecture as a `ForMultipleChoice` version in the library) on the SWAG dataset or your own data in a csv or a JSON file. The main difference is that this script exposes the bare training loop, to allow you to quickly experiment and add any customization you would like.

It offers less options than the script with `Trainer` (but you can easily change the options for the optimizer or the dataloaders directly in the script) but still run in a distributed setup, on TPU and supports mixed precision by the mean of the 🤗 `Accelerate` library. You can use the script normally after installing it:

```
pip install accelerate
```

then

```
export DATASET_NAME=swag

python run_swag_no_trainer.py \
  --model_name_or_path bert-base-cased \
  --dataset_name $DATASET_NAME \
  --max_seq_length 128 \
  --per_device_train_batch_size 32 \
  --learning_rate 2e-5 \
  --num_train_epochs 3 \
  --output_dir /tmp/$DATASET_NAME/
```

You can then use your usual launchers to run in it in a distributed environment, but the easiest way is to run

```
accelerate config
```

and reply to the questions asked. Then

```
accelerate test
```

that will check everything is ready for training. Finally, you can launch training with

```
export DATASET_NAME=swag

accelerate launch run_swag_no_trainer.py \
  --model_name_or_path bert-base-cased \
  --dataset_name $DATASET_NAME \
  --max_seq_length 128 \
  --per_device_train_batch_size 32 \
  --learning_rate 2e-5 \
  --num_train_epochs 3 \
  --output_dir /tmp/$DATASET_NAME/
```

This command is the same and will work for:

- a CPU-only setup
- a setup with one GPU
- a distributed training with several GPUs (single or multi node)
- a training on TPUs

Note that this library is in alpha release so your feedback is more than welcome if you encounter any problem using it.