Image Classification training examples

The following example showcases how to train/fine-tune `ViT` for image-classification using the JAX/Flax backend.

JAX/Flax allows you to trace pure functions and compile them into efficient, fused accelerator code on both GPU and TPU. Models written in JAX/Flax are **immutable** and updated in a purely functional way which enables simple and efficient model parallelism.

In this example we will train/fine-tune the model on the imagenette dataset.

Prepare the dataset

We will use the imagenette dataset to train/fine-tune our model. Imagenette is a subset of 10 easily classified classes from Imagenet (tench, English springer, cassette player, chain saw, church, French horn, garbage truck, gas pump, golf ball, parachute).

Download and extract the data.

```
wget https://s3.amazonaws.com/fast-ai-imageclas/imagenette2.tgz
tar -xvzf imagenette2.tgz
```

This will create a `imagenette2` dir with two subdirectories `train` and `val` each with multiple subdirectories per class. The training script expects the following directory structure

```
root/dog/xxx.png
root/dog/xxy.png
root/dog/[...]/xxz.png

root/cat/123.png
root/cat/nsdf3.png
root/cat/[...]/asd932_.png
```

Train the model

Next we can run the example script to fine-tune the model:

```
python run_image_classification.py \
    --output_dir ./vit-base-patch16-imagenette \
    --model_name_or_path google/vit-base-patch16-224-in21k \
    --train_dir="imagenette2/train" \
    --validation_dir="imagenette2/val" \
    --num_train_epochs 5 \
    --learning_rate 1e-3 \
    --per_device_train_batch_size 128 --per_device_eval_batch_size 128 \
    --overwrite_output_dir \
    --preprocessing_num_workers 32 \
    --push_to_hub
```

This should finish in ~7mins with 99% validation accuracy.