Fine-tuning a pretrained model

Agenda:

- How to prepare a large dataset from the Hub
- How to use the high-level `Trainer` API to fine-tune a model
- How to use a custom training loop
- How to leverage the 🤗 Accelerate library to easily run that custom training loop on any distributed setup

Processing the Data

1. Look at the code we are going to fine-tune a bert-base-uncased model on 2 sample sentence to see how the training process looks like.
2. Once we are done with this we move on to the task of Classifying. We use the MRPC dataset (Microsoft Research Paraphrase Corpus) dataset, introduced in a **paper** by William B. Dolan and Chris Brockett.

   The dataset consists of 5.8k pairs of sentences with the labels indicating if the paraphrases are same or not.

   This dataset is one of the 10 benchmarks in **GLUE benchmark** -General Language Understanding Evaluation- which is used to measure performance of ML models across 10 different text classification tasks

Datasets Library from Hugging Face

Check the code on module 3 to learn more about the Datasets capablities.

Preprocessing a dataset (sentence pairs in our case):

1. Tokenizer

There are 2 methods in which dataset can be preprocessed:

1. Directly passing the dataset to tokenizer class.
2. Applying a tokenizer on each example on the Dataset Object

We prefer method 2 since it returns a Dataset object as opposed to a dictnoary returned in the first step 1.

The next intresting idea we learn is about dynamic padding which is done using the DataCollatorWithPadding function. This function batches the dataset and also at the same time makes sure that the padding is max within a batch and not the max all over the dataset which traditionally is the scene.

2. Build a tf.data.Dataset object for train and validation.

3. Start training your KERAS model.