Fine tuning transformer:

2nd lab: Optional Pre-training LLMs with Hugging Face

**To install necessaries:**

# Install a specific version of the Transformers library

# Install the latest version of Transformers from the Hugging Face GitHub repository

# Install the datasets library (version 2.15.0 can be specified if required)

# Install the portalocker library with version >=2.0.0

# Install the latest version of Accelerate from the Hugging Face GitHub repository

# Install a specific version of PyTorch

# Upgrade and install the torchvision library

# Install a specific version of protobuf

**To import necessaries:**

Write Python code to set up a machine learning project using PyTorch, Hugging Face Transformers, and the Datasets library. Include the following functionalities:

* Neural network modeling and optimization
* Learning rate scheduling
* Tokenization and text preprocessing
* Dataset loading and collation for language modeling
* Training utilities, including argument setup and the Trainer API
* Progress tracking using TQDM
* Handling paths, timing, and mathematical computations
* Suppressing warnings to improve code readability

Ensure all necessary imports are included, with modules for training models such as BERT and utilities like AdamW optimizers and learning rate schedulers. Don’t do anything except import.

**Exercise:**  
1. Create a model and tokenizer using Hugging Face library.

2. Go to this [link](https://huggingface.co/datasets?task\_categories=task\_categories:text-classification&sort=trending)

3. Choose a Text Classification dataset that you can load, for instance 'stanfordnlp/snli'

4. Use that dataset to train your model(please be mindful of the resources available for the training) and evaluate it.

Answer:

from transformers import AutoTokenizer, AutoModelForSequenceClassification

from datasets import load\_dataset

#Dataset has 3 label class and three columns(last one for labelling)

from transformers import TrainingArguments, Trainer

from sklearn.metrics import accuracy\_score