```
Start coding or generate with AI.
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

**First of all let's clone the longformer repository and nevigate to its directory**

```
!git clone https://github.com/allenai/longformer.git
```

```
Cloning into 'longformer'...
remote: Enumerating objects: 1240, done.
remote: Counting objects: 100% (215/215), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 1240 (delta 203), reused 198 (delta 198), pack-reused 1025
Receiving objects: 100% (1240/1240), 837.38 KiB | 1.23 MiB/s, done.
Resolving deltas: 100% (838/838), done.
```

```
%cd longformer
```

```
/content/longformer
```

**Now we will install require packages**

```
!pip install transformers datasets
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.42.4)
Collecting datasets
  Downloading datasets-2.20.0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.15.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.23.5)
Requirement already satisfied: numpy<2.0,>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2024.5.15)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.4)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.5)
Collecting pyarrow>=15.0.0 (from datasets)
  Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl.metadata (3.3 kB)
Requirement already satisfied: pyarrow-hotfix in /usr/local/lib/python3.10/dist-packages (from datasets) (0.6)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.1.4)
Collecting xxhash (from datasets)
  Downloading xxhash-3.4.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess (from datasets)
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Collecting fsspec<=2024.5.0,>=2023.1.0 (from fsspec[http]<=2024.5.0,>=2023.1.0->datasets)
  Downloading fsspec-2024.5.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.10.1)
```

```
    Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.3.4)
    Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
    Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (24.2.0)
    Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
    Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.0.5)
    Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.4)
    Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
    Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.2->transformers) (4.12.2)
    Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.3.2)
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.7)
    Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7)
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.7.4)
    Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
    Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
    Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
    Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->datasets) (1.16.0)
    Downloading datasets-2.20.0-py3-none-any.whl (547 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 547.8/547.8 kB 31.8 MB/s eta 0:00:00
    Downloading dill-0.3.8-py3-none-any.whl (116 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 11.0 MB/s eta 0:00:00
    Downloading fsspec-2024.5.0-py3-none-any.whl (316 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 316.1/316.1 kB 23.1 MB/s eta 0:00:00
    Downloading pyarrow-17.0.0-cp310-cp310-manylinux_2_28_x86_64.whl (39.9 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 39.9/39.9 MB 16.9 MB/s eta 0:00:00
    Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 11.5 MB/s eta 0:00:00
    Downloading xxhash-3.4.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 14.1 MB/s eta 0:00:00
    Installing collected packages: xxhash, pyarrow, fsspec, dill, multiprocess, datasets
      Attempting uninstall: pyarrow
```

### Lets import all the necessary libraries

```
import torch
from transformers import LongformerTokenizer, LongformerModel
```

### loading pretrained longformer

```
model_name = 'allenai/longformer-base-4096'
tokenizer = LongformerTokenizer.from_pretrained(model_name)
model = LongformerModel.from_pretrained(model_name)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
vocab.json:  100%                                  899k/899k  [00:00<00:00,  2.00MB/s]
merges.txt:  100%                                  456k/456k  [00:00<00:00,  1.07MB/s]
tokenizer.json:  100%                                1.36M/1.36M  [00:00<00:00,  21.2MB/s]
config.json:  100%                                 694/694  [00:00<00:00,  12.2kB/s]
pytorch_model.bin:  100%                                597M/597M  [00:01<00:00,  283MB/s]
```

**now we will prepare input and run the model**

```
text = "This is an example of a very long document that we want to process using Longformer."
inputs = tokenizer(text, return_tensors='pt', max_length=4096, truncation=True, padding='max_length')
```

```
with torch.no_grad():
    outputs = model(**inputs)

# Retrieve the last hidden state
last_hidden_states = outputs.last_hidden_state


# Print last hidden state details
print("Last Hidden States Shape:", last_hidden_states.shape)
print("Last Hidden States:", last_hidden_states)
```

```
Last Hidden States Shape: torch.Size([1, 4096, 768])
Last Hidden States: tensor([[[-0.0470,  0.1103, -0.0197,  ..., -0.1658,  0.0115, -0.0037],
         [-0.0020,  0.1571,  0.1532,  ...,  0.0158,  0.1495,  0.2231],
         [ 0.1668,  0.3675,  0.1867,  ..., -0.3038,  0.1431,  0.3824],
         ...,
         [-0.0236,  0.0741, -0.0145,  ..., -0.0990, -0.0409, -0.0745],
         [-0.0236,  0.0741, -0.0145,  ..., -0.0990, -0.0409, -0.0745],
         [-0.0236,  0.0741, -0.0145,  ..., -0.0990, -0.0409, -0.0745]]])
```

```
print("Tokenized Input IDs:", inputs['input_ids'])
print("Attention Mask:", inputs['attention_mask'])
```

```
Tokenized Input IDs: tensor([[  0, 713,  16,  ...,   1,   1,   1]])
Attention Mask: tensor([[1, 1, 1,  ..., 0, 0, 0]])
```

**Now lets create an upgraded version of the existing methodology.**

**Lets load a dataset for finetuning , here we have used IMDb dataset for sentiment analysis.**

```
from datasets import load_dataset

dataset = load_dataset('imdb')
```

```
Downloading readme: 100%                          7.81k/7.81k [00:00<00:00, 195kB/s]
Downloading data: 100%                            21.0M/21.0M [00:00<00:00, 14.4MB/s]
Downloading data: 100%                            20.5M/20.5M [00:00<00:00, 18.7MB/s]
Downloading data: 100%                            42.0M/42.0M [00:00<00:00, 78.1MB/s]
Generating train split: 100%                      25000/25000 [00:00<00:00, 95944.42 examples/s]
Generating test split: 100%                       25000/25000 [00:00<00:00, 111560.81 examples/s]
Generating unsupervised split: 100%                  50000/50000 [00:00<00:00, 86069.19 examples/s]
```

**Modify Model Hyperparameters**

```
import torch
from transformers import LongformerTokenizer, LongformerForSequenceClassification, Trainer, TrainingArguments

model_name = 'allenai/longformer-base-4096'
tokenizer = LongformerTokenizer.from_pretrained(model_name)
model = LongformerForSequenceClassification.from_pretrained(model_name, num_labels=2)
```

```
Some weights of LongformerForSequenceClassification were not initialized from the model checkpoint at allenai/longformer-base-4096 and are newly initialized: ['classifier.dense
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
def tokenize_function(examples):
    return tokenizer(examples['text'], padding="max_length", truncation=True, max_length=512)

tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

```
Map: 100%                 25000/25000 [01:34<00:00, 417.33 examples/s]
Map: 100%                 25000/25000 [00:58<00:00, 582.71 examples/s]
Map: 100%                 50000/50000 [01:30<00:00, 602.98 examples/s]
```

**Prepare Data for Training:**

```
tokenized_datasets = tokenized_datasets.remove_columns(['text'])
tokenized_datasets = tokenized_datasets.rename_column("label", "labels")
tokenized_datasets.set_format("torch")

train_dataset = tokenized_datasets["train"].shuffle(seed=42).select(range(1000))  # Subset for quick experimentation
eval_dataset = tokenized_datasets["test"].shuffle(seed=42).select(range(500))
```

**Experiment with Hyperparameters ,We'll modify the learning rate and batch size as examples of hyperparameter changes.**

**Set Training Arguments:**

```
training_args = TrainingArguments(
    output_dir="./results",
    evaluation_strategy="epoch",
    learning_rate=3e-5,          # Modified learning rate
    per_device_train_batch_size=8,  # Modified batch size
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    weight_decay=0.01,
)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1494: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transf
  warnings.warn(
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
)

trainer.train()
```

```
Initializing global attention on CLS token...
```

[375/375 12:06, Epoch 3/3]

| Epoch | Training Loss | Validation Loss |
|---|---|---|
| 1 | No log | 0.285210 |
| 2 | No log | 0.361324 |
| 3 | No log | 0.480662 |

```
TrainOutput(global_step=375, training_loss=0.3153460896809896, metrics={'train_runtime': 730.4581, 'train_samples_per_second': 4.107, 'train_steps_per_second': 0.513,
```

**Evaluate on the Test Set:**

```
eval_results = trainer.evaluate()
print(eval_results)
```

```
[63/63 00:25]
{'eval_loss': 0.4896616900351379, 'eval_runtime': 26.0744, 'eval_samples_per_second': 19.176, 'eval_steps_per_second': 2.416, 'epoch': 3.0}
```

**Let's start by setting up a baseline model with default parameters and then we will change each parameter to see the differences.**

```
from transformers import TrainingArguments, Trainer

# Set up baseline training arguments
baseline_training_args = TrainingArguments(
    output_dir="./results_baseline",
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    weight_decay=0.01,
)

# Initialize the Trainer with baseline settings
baseline_trainer = Trainer(
    model=model,
    args=baseline_training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
)

# Train the baseline model
baseline_trainer.train()

# Evaluate the baseline model
baseline_eval_results = baseline_trainer.evaluate()
print("Baseline Evaluation Results:", baseline_eval_results)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1494: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transf
  warnings.warn(
[302/375 09:19 < 02:16, 0.54 it/s, Epoch 2.41/3]
```

| Epoch | Training Loss | Validation Loss |
|---|---|---|
| 1 | No log | 0.554120 |
| 2 | No log | 0.522093 |

```
[375/375 12:03, Epoch 3/3]
```

| Epoch | Training Loss | Validation Loss |
|---|---|---|
| 1 | No log | 0.554120 |
| 2 | No log | 0.522093 |
| 3 | No log | 0.597216 |

```
[63/63 00:25]
Baseline Evaluation Results: {'eval_loss': 0.5972161293029785, 'eval_runtime': 25.9862, 'eval_samples_per_second': 19.241, 'eval_steps_per_second': 2.424, 'epoch': 3.0}
```

**Now We'll test three different learning rates: 1e-5, 2e-5, and 5e-5.**

```
learning_rates = [1e-5, 2e-5, 5e-5]

for lr in learning_rates:
    print(f"Training with learning rate: {lr}")

    training_args = TrainingArguments(
        output_dir=f"./results_lr_{lr}",
        evaluation_strategy="epoch",
        learning_rate=lr,
        per_device_train_batch_size=8,
        per_device_eval_batch_size=8,
        num_train_epochs=3,
        weight_decay=0.01,
    )

    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=train_dataset,
        eval_dataset=eval_dataset,
    )

    # Train the model with the specified learning rate
    trainer.train()

    # Evaluate the model
    eval_results = trainer.evaluate()
    print(f"Evaluation Results for learning rate {lr}:", eval_results)
```

```
Training with learning rate: 1e-05
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1494: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transf
  warnings.warn(
```

[375/375 12:04, Epoch 3/3]

| Epoch | Training Loss | Validation Loss |
|---|---|---|
| 1 | No log | 0.613347 |
| 2 | No log | 0.753174 |
| 3 | No log | 0.633946 |

[63/63 00:25]

```
Evaluation Results for learning rate 1e-05: {'eval_loss': 0.6339464783668518, 'eval_runtime': 26.0984, 'eval_samples_per_second': 19.158, 'eval_steps_per_second': 2.414, 'epoc
Training with learning rate: 2e-05
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1494: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transf
  warnings.warn(
```

[375/375 12:28, Epoch 3/3]

| Epoch | Training Loss | Validation Loss |
|---|---|---|
| 1 | No log | 0.680033 |
| 2 | No log | 0.812165 |
| 3 | No log | 0.692521 |

[63/63 00:25]

```
Evaluation Results for learning rate 2e-05: {'eval_loss': 0.6925205588340759, 'eval_runtime': 26.1084, 'eval_samples_per_second': 19.151, 'eval_steps_per_second': 2.413, 'epoc
Training with learning rate: 5e-05
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1494: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transf
  warnings.warn(
```

[375/375 12:18, Epoch 3/3]

| Epoch | Training Loss | Validation Loss |
|---|---|---|
| 1 | No log | 0.914951 |
| 2 | No log | 0.791469 |
| 3 | No log | 0.751127 |

[63/63 00:25]

```
Evaluation Results for learning rate 5e-05: {'eval_loss': 0.751127302646637, 'eval_runtime': 26.0991, 'eval_samples_per_second': 19.158, 'eval_steps_per_second': 2.414, 'epoch
```

**Also Let's experiment with batch sizes of 4**

```
# Experiment with Batch Size 4
batch_size = 4
print(f"Training with batch size: {batch_size}")

training_args_bs4 = TrainingArguments(
    output_dir=f"./results_bs_{batch_size}",
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    weight_decay=0.01,
)

trainer_bs4 = Trainer(
    model=model,
    args=training_args_bs4,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
)

# Train the model with batch size 4
trainer_bs4.train()

# Evaluate the model
eval_results_bs4 = trainer_bs4.evaluate()
print(f"Evaluation Results for batch size {batch_size}:", eval_results_bs4)
```

```
Training with batch size: 4
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1494: FutureWarning: `evaluation_strategy` is deprecated and will be removed in version 4.46 of 🤗 Transf
  warnings.warn(
```

[750/750 13:19, Epoch 3/3]

| Epoch | Training Loss | Validation Loss |
|---|---|---|
| 1 | No log | 1.269984 |
| 2 | 0.047400 | 1.193018 |
| 3 | 0.047400 | 1.204575 |

[125/125 00:27]

```
Evaluation Results for batch size 4: {'eval_loss': 1.204574704170227, 'eval_runtime': 28.2237, 'eval_samples_per_second': 17.716, 'eval_steps_per_second': 4.429, 'epoch': 3.0}
```

**now as a final experiment We will try training the model for 2,and 4 epochs.**

```
epochs = [2,4]

for epoch in epochs:
    print(f"Training for {epoch} epochs")

    training_args = TrainingArguments(
        output_dir=f"./results_epochs_{epoch}",
        evaluation_strategy="epoch",
        learning_rate=2e-5,
        per_device_train_batch_size=8,
        per_device_eval_batch_size=8,
        num_train_epochs=epoch,
        weight_decay=0.01,
    )

    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=train_dataset,
```