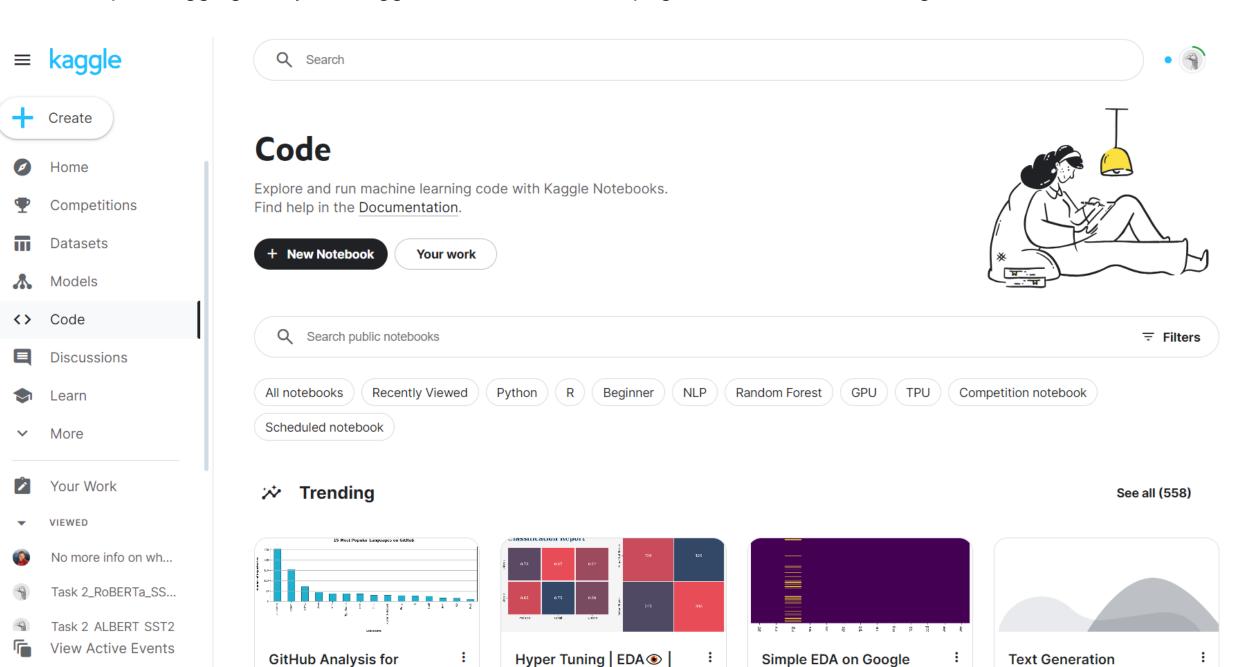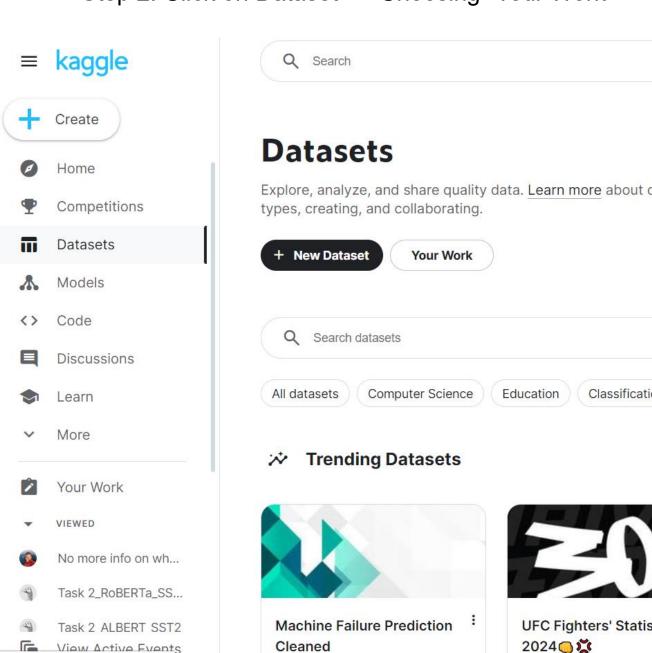# Tutorial on how to train BERT with GPU on Kaggle

Nguyen Xuan Binh

Nguyen Long

Step 1: Logging into your Kaggle account. The main page should look something like this
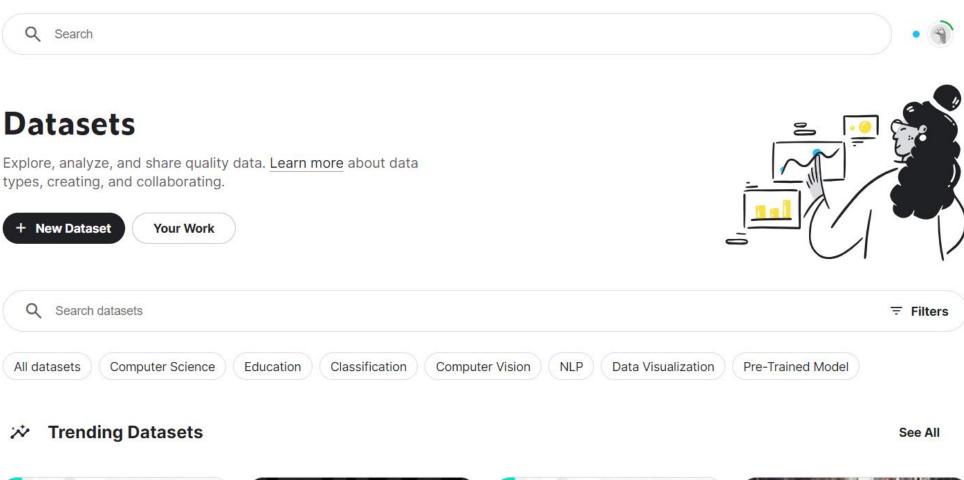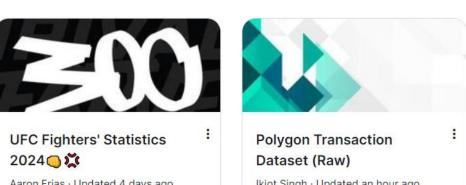
Step 2: Click on Dataset => Choosing "Your Work"

kaggle

🔍 Search

+ Create

🧭 Home

🏆 Competitions

☰ Datasets

⚛ Models

<> Code

💬 Discussions

🎓 Learn

⌄ More

📝 Your Work

⌄ VIEWED

👤 No more info on wh...

👤 Task 2_RoBERTa_SS...

👤 Task 2_ALBERT_SST2

📑 View Active Events

Datasets

Explore, analyze, and share quality data. Learn more about data
types, creating, and collaborating.

[+ New Dataset] [Your Work]

🔍 Search datasets   ☰ Filters

All datasets   Computer Science   Education   Classification   Computer Vision   NLP   Data Visualization   Pre-Trained Model

📈 **Trending Datasets**   See All

Machine Failure Prediction Cleaned

Ikiot Singh · Updated a day ago

UFC Fighters' Statistics 2024🥊💥

Aaron Frias · Updated 4 days ago

Polygon Transaction Dataset (Raw)

Ikiot Singh · Updated an hour ago

Taxi Trips Chicago 2024

Adel Anseur · Updated 10 days ago

Usability **9.4** · 116 MB
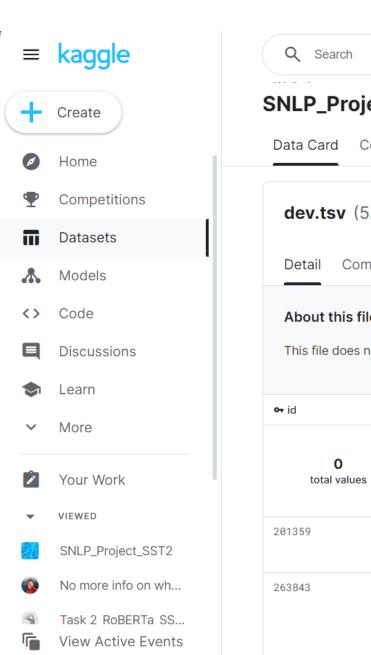
Step 3: Create New Dataset and upload QQP and SST2 dataset

SST2 dataset contains dev.tsv, test.tsv and train.tsv

kaggle

Q Search

SNLP_Project_SST2   ▲ 0   New Notebook   ⬇ Download (8 MB) ▾   ⋮

+ Create

Data Card   Code (0)   Discussion (0)   Suggestions (0)   Settings

⊘ Home

🏆 Competitions

▥ Datasets   **dev.tsv** (94.93 kB)   ⬇ ⟦ ⟧ >

⚙ Models

<> Code   Detail   Compact   Column   2 of 2 columns ∨

▤ Discussions

🎓 Learn   **About this file**   ✎

                          This file does not have a description yet.

✏ Your Work

∨ VIEWED

   No more info on wh...

   Task 2_RoBERTa_SS...

   Task 2 ALBERT SST2
🗗 View Active Events

Data Explorer

Version 1 (24.48 MB)

▸ ■ original
  ‖‖ dev.tsv
  ‖‖ test.tsv
  ‖‖ train.tsv

� sentence	# label
868 unique values	**391** total values
it 's a charming and often affecting journey .	1
unflinchingly bleak and desperate	0
allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmma...	1

Summary

QQP dataset contains dev.tsv, test.tsv and train.tsv

Step 4: Click on Code tab => "Your work"

Step 5: Click on Create => "New notebook"

Step 6: Click on Import Notebook => Choose our ipynb developed notebook

notebook2fe06ebc45 *Draft saved*

File   Edit   View   Run   Add-ons   Help

+	New Notebook
⬆	Import Notebook
⬇	Download Notebook
	Upgrade to Google Cloud AI Notebooks
	Open in Colab
	Set as Utility Script
🏛	Add input
	Upload input ▸
✏	Language ▸
<>	Editor Type ▸
👥	Share
	Link to GitHub
⟳	Save Version
	Version history
🗑	Delete Notebook

👥 Share   Save Version   0

● Draft Session off (run a cell to start)

comes with many helpful analytics libraries installed
e/python Docker image: https://github.com/kaggle/docker-python
l helpful packages to load

algebra
rocessing, CSV file I/O (e.g. pd.read_csv)

able in the read-only "../input/" directory
(by clicking run or pressing Shift+Enter) will list all files under the

os.walk('/kaggle/input'):
s:
irname, filename))

to the current directory (/kaggle/working/) that gets preserved as output
ry files to /kaggle/temp/, but they won't be saved outside of the curren

Notebook

Input   ∧

＋ Add Input   ⬆ Upload

No input attached

Attach a Kaggle dataset, model, or competition

Output   ∧

▸ 📁 /kaggle/working

Session options   ∧

ACCELERATOR

Step 6: Now your notebook should look like this. Then click on "Add input"

Task 2_DeBERTa_SST2

File   Edit   View   Run   Add-ons   Help

Share   **Save Version** 1

Markdown ▾   ● Draft Session off (run a cell to start)

Fine-Tuning BERT on Stanford Sentiment Treebank 2

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT, DeBERTa) with the HuggingFace PyTorch library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transformers)

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dataset

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from movie reviews and human annotations of their sentiment. The task is to determine whether a given sentence has positive or negative sentiment.

Data fields

- sentence - the review sentence whose sentiment should be determined
- label - 0 for negative sentiment, 1 for positive sentiment

Notebook

Input ⌃

+ Add Input   ⬆ Upload

No input attached
Attach a Kaggle dataset, model, or competition

Output ⌃

▸ 📁 /kaggle/working

Session options ⌃

ACCELERATOR

Step 7: Click on "Your work" and choose either QQP or SST2 dataset, depending on the task

Task 2_DeBERTa_SST2   Draft saved

File   Edit   View   Run   Add-ons   Help

Share   Save Version   1

+   ✂   📋   📋   ▶   ▶▶   Run All   Markdown ▾   ● Draft Session off (run a cell to start)   ⏻   🗘   ⋮

Fine-Tuning BERT on Stanford Sentiment Treebank 2

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT, DeBERTa) with the HuggingFace PyTorch library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transformers)

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dataset

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from movie reviews and human annotations of their sentiment. The task is to determine whether a given sentence has positive or negative sentiment.

Data fields

- sentence - the review sentence whose sentiment should be determined
- label - 0 for negative sentiment, 1 for positive sentiment

Add Input   ✕

🔍 Search keyword or URL

All   Your Work   Datasets   Models

Competition Datasets   Notebook

4 Recently Viewed Datasets

SNLP_Project_QQP
Spring Nuance · Updated 3d ago
0 Upvotes · 4 Files (other) · 64 MB   ⊕

SNLP_Project_SST2
Spring Nuance · Updated 2d ago
0 Upvotes · 11 Files (other) · 8 MB   ⊕

Stanford Sentiment Treebank v2...
Atul Anand {Jha} · Updated 4y ago
68 Upvotes · 16 Files (other) · 20 MB   ⊕

Mushroom Classification
UCI Machine Learning · Updated 7y ago
2331 Upvotes · 1 File (CSV) · 35 kB   ⊕

Step 7: Now the dataset has been loaded into your Input tab

Task 2_DeBERTa_SST2

File   Edit   View   Run   Add-ons   Help

▶   ▶▶   Run All   Markdown ▾   ● Draft Session off (run a cell to start)

Notebook

Input

[+ Add Input]   [↥ Upload]

DATASETS
▸ 🔷 snlp-project-sst2

Fine-Tuning BERT on Stanford Sentiment Treebank 2

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT, DeBERTa) with the HuggingFace PyTorch library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transformers)

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dataset

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from movie reviews and human annotations of their sentiment. The task is to determine whether a given sentence has positive or negative sentiment.

Data fields

- sentence - the review sentence whose sentiment should be determined
- label - 0 for negative sentiment, 1 for positive sentiment

Output

▸ ■ /kaggle/working

Session options

ACCELERATOR

GPU T4 ×2 ▾

Quota: 25:02 / 30 hrs

LANGUAGE

Python ▾

PERSISTENCE

No persistence ▾

Step 8: In Session options, choose Accelerator: GPU T4 x 2, Language: Python

Task 2_DeBERTa_SST2   Draft saved

File   Edit   View   Run   Add-ons   Help

Share   Save Version   1

+   ✂   ▤   ▤   ▶   ▶▶   Run All   Markdown ▾   ● Draft Session off (run a cell to start)   ⏻   ↻   ⋮

▶   📁 /kaggle/working

Fine-Tuning BERT on Stanford Sentiment Treebank 2

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT, DeBERTa) with the HuggingFace PyTorch library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transformers)

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dataset

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from movie reviews and human annotations of their sentiment. The task is to determine whether a given sentence has positive or negative sentiment.

Data fields

- sentence - the review sentence whose sentiment should be determined
- label - 0 for negative sentiment, 1 for positive sentiment

Session options ∧

ACCELERATOR

GPU T4 ×2 ▾

Quota: 25:02 / 30 hrs

LANGUAGE

Python ▾

PERSISTENCE

Files only ▾

ENVIRONMENT

Always use latest environment ▾

Always get the latest package versions, but you may have to modify your code. What is a notebook environment? ⓘ

INTERNET

⬤ Internet on

TAGS

GPU ✎

Step 8: In Session options, choose Persistence: Files only, and Environment: Always use latest environment

Task 2_DeBERTa_SST2 Draft saved

File Edit View Run Add-ons Help

Share | Save Version 1

Markdown ▼ ● Draft Session off (run a cell to start) Run All

Fine-Tuning BERT on Stanford Sentiment Treebank 2

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT, DeBERTa) with the HuggingFace PyTorch library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transformers)

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dataset

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from movie reviews and human annotations of their sentiment. The task is to determine whether a given sentence has positive or negative sentiment.

Data fields

- sentence - the review sentence whose sentiment should be determined
- label - 0 for negative sentiment, 1 for positive sentiment

▶ ▣ /kaggle/working

Session options ⌃

ACCELERATOR
GPU T4 ×2 ▼

Quota: 25:02 / 30 hrs

LANGUAGE
Python ▼

PERSISTENCE
Files only ▼

ENVIRONMENT
Always use latest environment ▼

Always get the latest package versions, but you may have to modify your code. What is a notebook environment? ⓘ
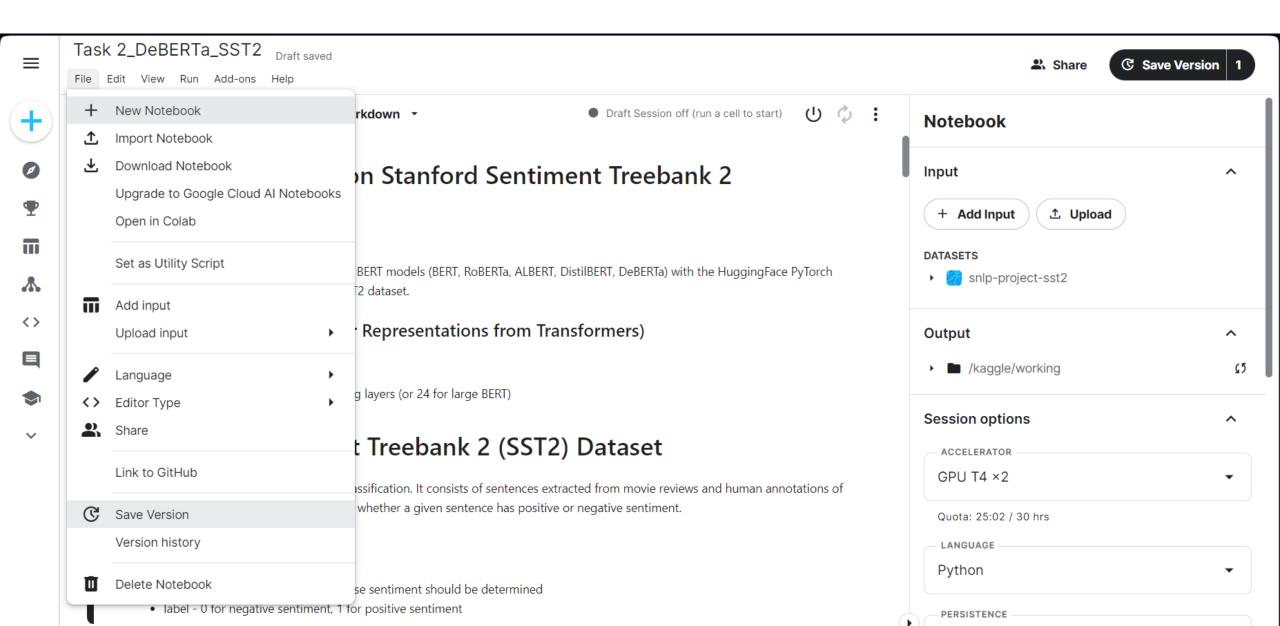
INTERNET
● Internet on

TAGS
GPU ✎

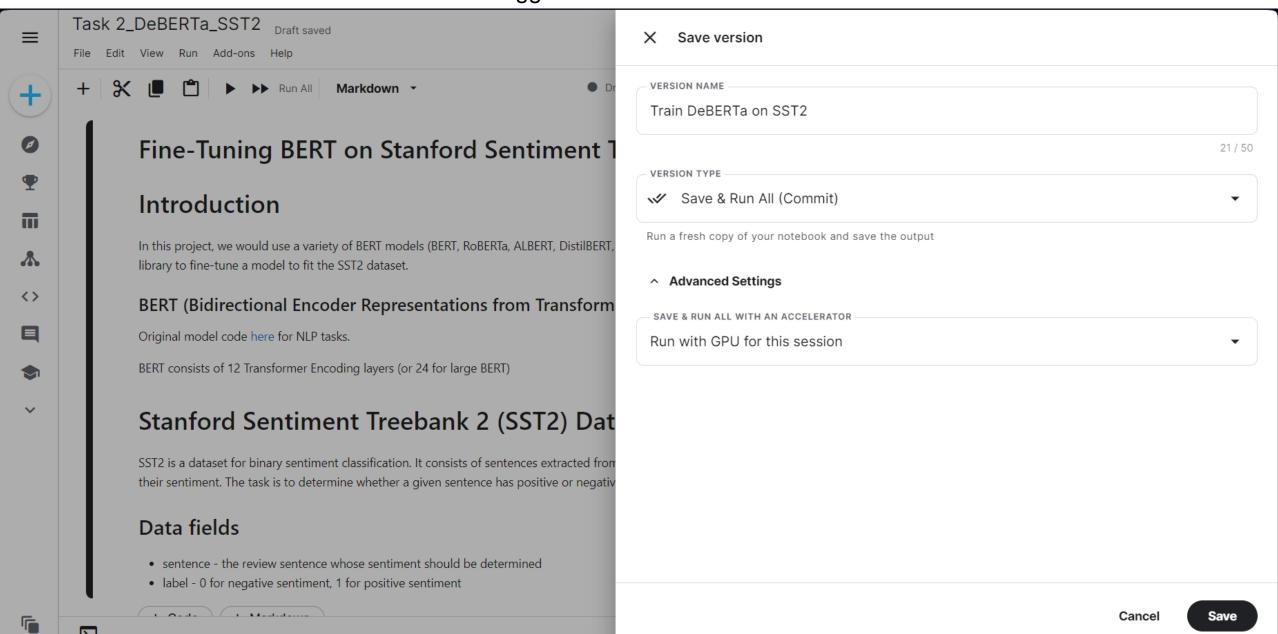**Step 8: Toggle on the Internet On. If you don't do this, BERT cannot download the pretrained model**

Step 9: Click Run All to train the BERT model from start to finish. The results would appear in kaggle/working

Task 2_DeBERTa_SST2 Draft saved

File   Edit   View   Run   Add-ons   Help

Share   Save Version 1

Run All   Markdown ▾   Draft Session off (run a cell to start)

Run all

Fine-Tuning BERT on Stanford Sentiment Treebank 2

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT, DeBERTa) with the HuggingFace PyTorch library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transformers)

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dataset

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from movie reviews and human annotations of their sentiment. The task is to determine whether a given sentence has positive or negative sentiment.

Data fields

- sentence - the review sentence whose sentiment should be determined
- label - 0 for negative sentiment, 1 for positive sentiment

Notebook

Input   ⌃

+ Add Input   ⬆ Upload

DATASETS

▸ 🔵 snlp-project-sst2

Output   ⌃

▸ 📁 /kaggle/working

Session options   ⌃

ACCELERATOR

GPU T4 ×2

Quota: 25:02 / 30 hrs

LANGUAGE
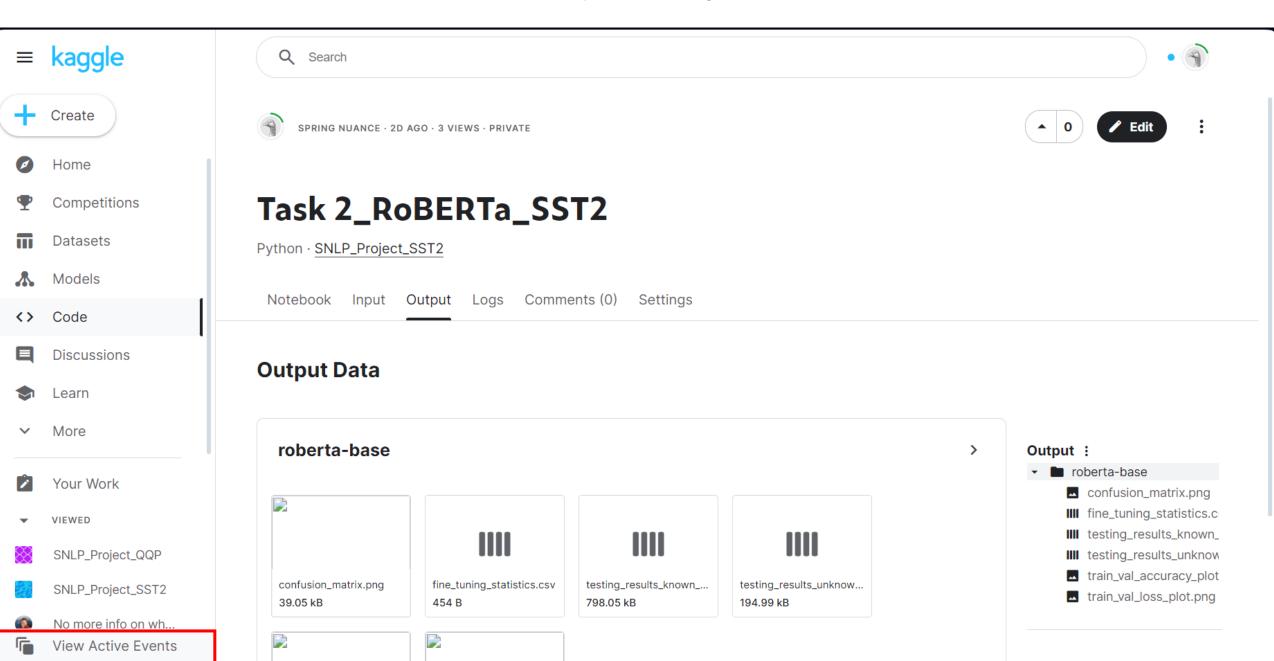
Python

PERSISTENCE

Files only

Step 10: When we train models, it is likely that we want to let it run overnight and we cannot keep the computer opened forever. To do this, we can choose File > Save Version.
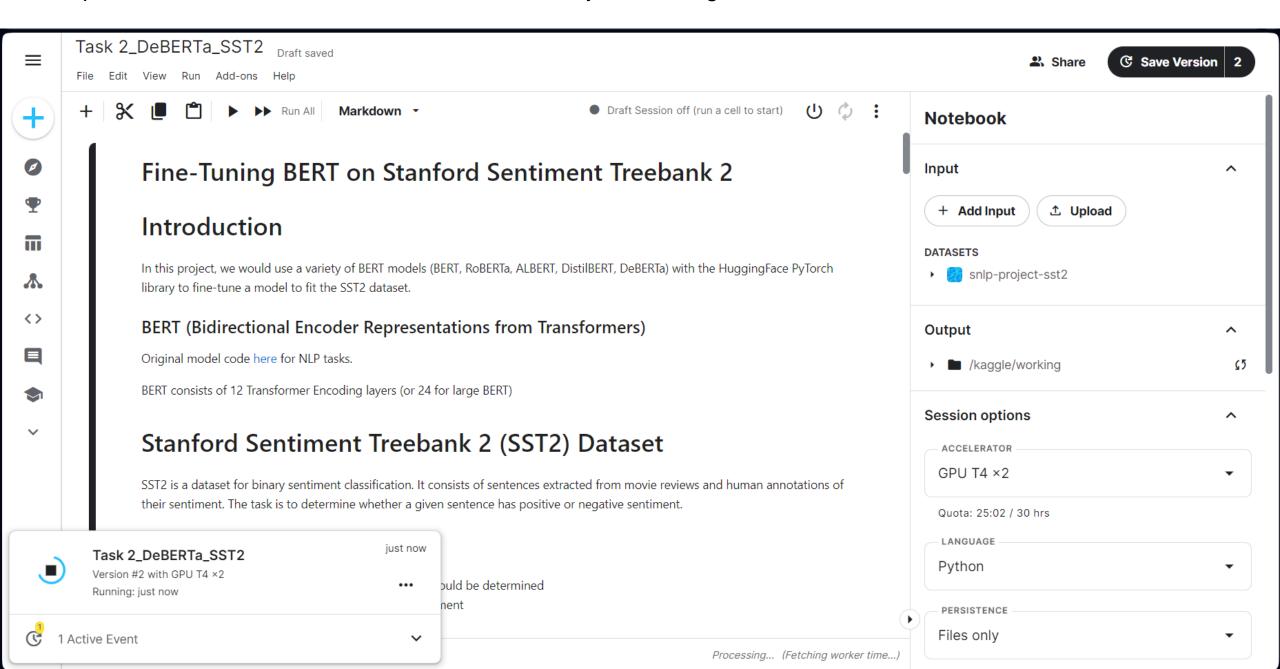
Step 10: Name the version, choose version type as Save & Run all, and choose Run with GPU for this session
The notebook would be submitted to the Kaggle server and runs even when we close the browser.

Task 2_DeBERTa_SST2   Draft saved

File   Edit   View   Run   Add-ons   Help

+   ✂   ▣   ▢   ▶   ▶▶   Run All | Markdown ▾   ● Dr

Fine-Tuning BERT on Stanford Sentiment T

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT,
library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transform

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dat

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from
their sentiment. The task is to determine whether a given sentence has positive or negativ

Data fields

- sentence - the review sentence whose sentiment should be determined
- label - 0 for negative sentiment, 1 for positive sentiment

✕   **Save version**

VERSION NAME

Train DeBERTa on SST2

21 / 50

VERSION TYPE

✅   Save & Run All (Commit)   ▾

Run a fresh copy of your notebook and save the output

∧   **Advanced Settings**

SAVE & RUN ALL WITH AN ACCELERATOR

Run with GPU for this session   ▾
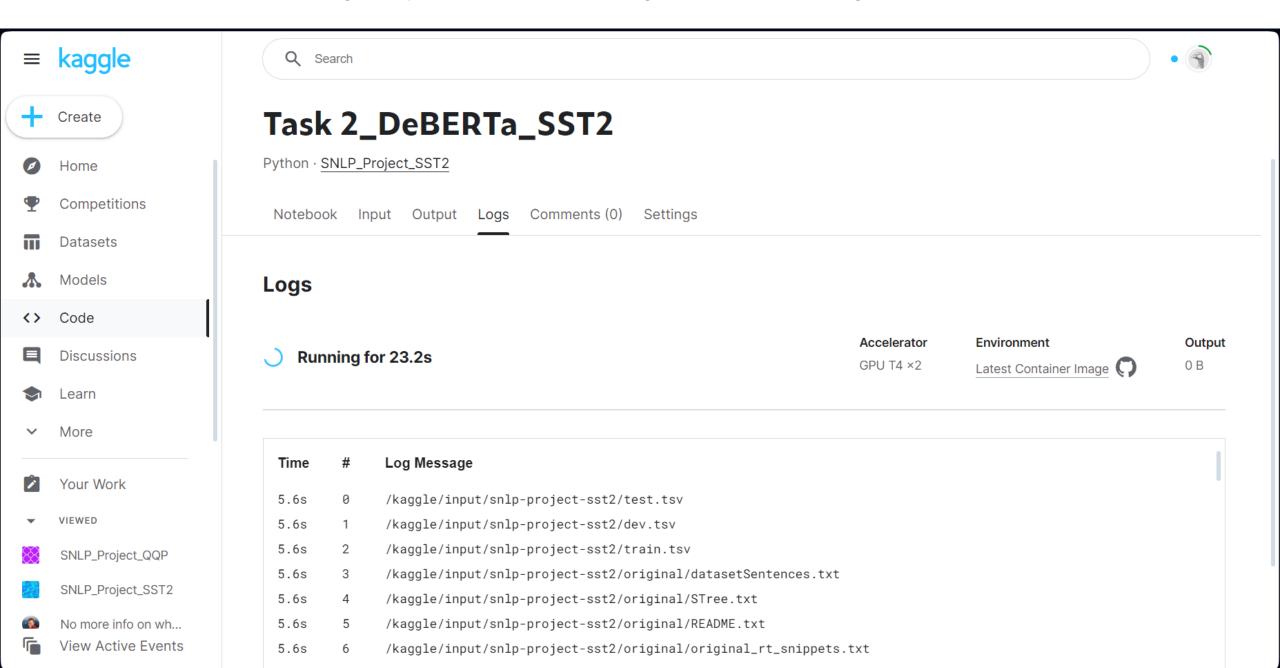
Cancel   **Save**

Step 11: Click on View Active Events. You can find your running GPU session

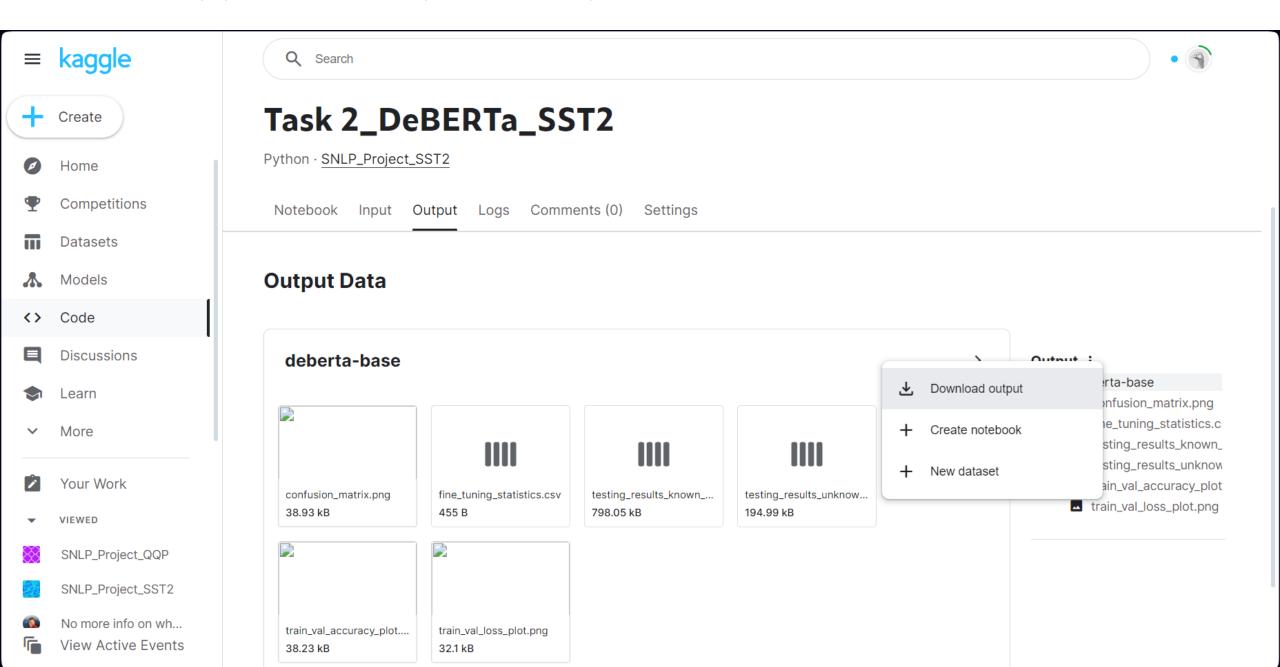# Step 11: Click on View Active Events. You can find your running GPU session

Task 2_DeBERTa_SST2 Draft saved

File Edit View Run Add-ons Help

Share Save Version 2

+ ✂ ▣ ▢ ▶ ▶▶ Run All Markdown ▾   ● Draft Session off (run a cell to start) ⏻ ↻ ⋮

Fine-Tuning BERT on Stanford Sentiment Treebank 2

Introduction

In this project, we would use a variety of BERT models (BERT, RoBERTa, ALBERT, DistilBERT, DeBERTa) with the HuggingFace PyTorch library to fine-tune a model to fit the SST2 dataset.

BERT (Bidirectional Encoder Representations from Transformers)

Original model code here for NLP tasks.

BERT consists of 12 Transformer Encoding layers (or 24 for large BERT)

Stanford Sentiment Treebank 2 (SST2) Dataset

SST2 is a dataset for binary sentiment classification. It consists of sentences extracted from movie reviews and human annotations of their sentiment. The task is to determine whether a given sentence has positive or negative sentiment.

Notebook

Input ⌃

+ Add Input   ⬆ Upload

DATASETS

▸ 🔷 snlp-project-sst2

Output ⌃

▸ ▰ /kaggle/working   ⟲

Session options ⌃

ACCELERATOR

GPU T4 ×2 ▾

Quota: 25:02 / 30 hrs

LANGUAGE

Python ▾

PERSISTENCE

Files only ▾

Task 2_DeBERTa_SST2   just now
Version #2 with GPU T4 ×2
Running: just now   •••

⏱¹ 1 Active Event   ⌄

...uld be determined
...ent

Processing... (Fetching worker time...)

Step 12: Click on the running job, you can look at the Logs to see the running process.

Step 13: After training the model successfully, it would return output files in Output tab.

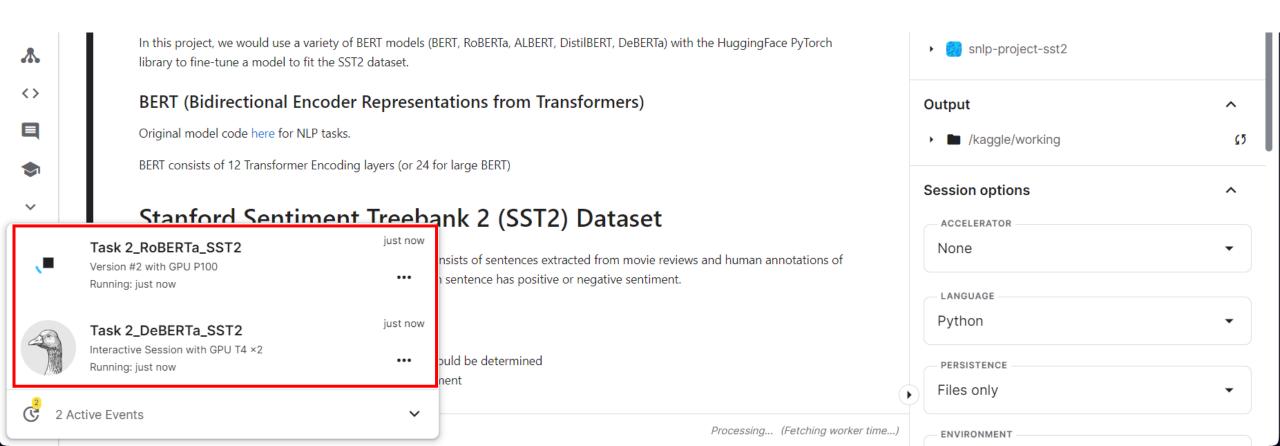# Step 13: Finally, you can download your outputs to your local computer
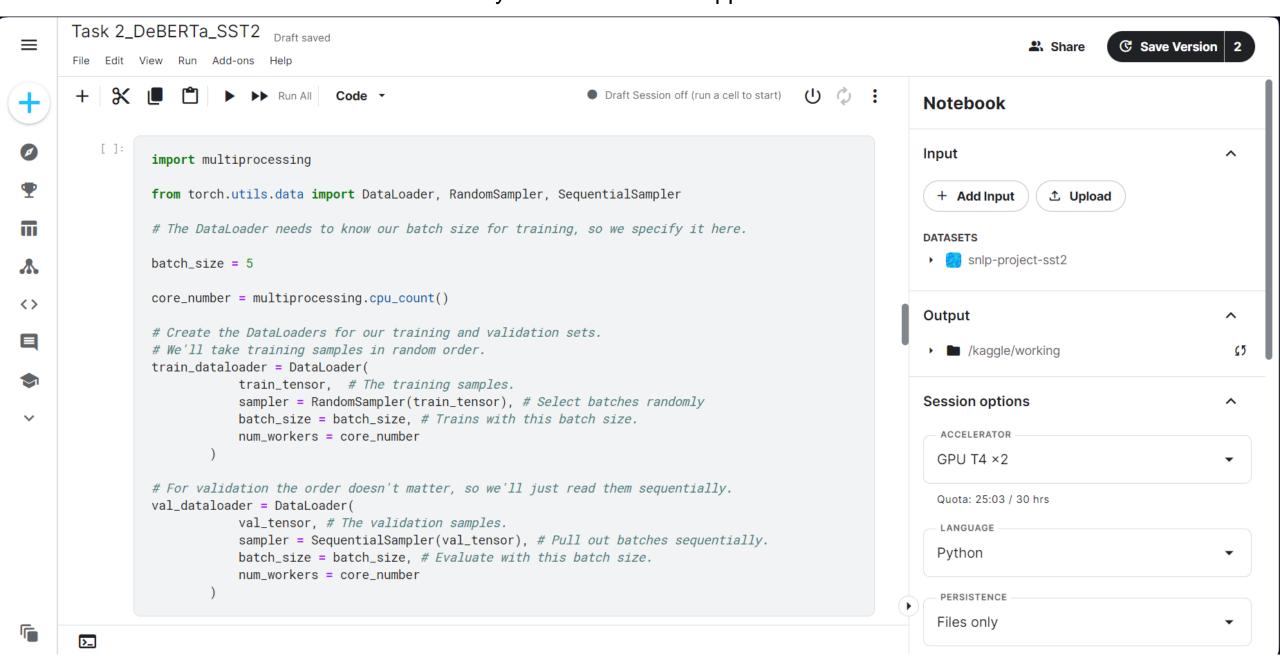
Step 14: Note that there is a limit on the number of GPU active sessions and interactive session

Saved versions using GPU: at most 2 GPU sessions
Interactive session using GPU: at most 1 GPU session

Saved versions is just the tutorial step we follow above. Interactive session is we run the notebook directly on the browser and it would be interrupted if we close the browser.

Finally, you should use GPU T4. Using other GPU type would result in running out of memory for GPU, no matter how small the batch size. GPU T4 is the only version that can support batch size of 5-10 sentences.

Task 2_DeBERTa_SST2 Draft saved

File   Edit   View   Run   Add-ons   Help

Run All   Code ▾   ● Draft Session off (run a cell to start)

[ ]:
```python
import multiprocessing

from torch.utils.data import DataLoader, RandomSampler, SequentialSampler

# The DataLoader needs to know our batch size for training, so we specify it here.

batch_size = 5

core_number = multiprocessing.cpu_count()

# Create the DataLoaders for our training and validation sets.
# We'll take training samples in random order.
train_dataloader = DataLoader(
            train_tensor,  # The training samples.
            sampler = RandomSampler(train_tensor), # Select batches randomly
            batch_size = batch_size, # Trains with this batch size.
            num_workers = core_number
        )

# For validation the order doesn't matter, so we'll just read them sequentially.
val_dataloader = DataLoader(
            val_tensor, # The validation samples.
            sampler = SequentialSampler(val_tensor), # Pull out batches sequentially.
            batch_size = batch_size, # Evaluate with this batch size.
            num_workers = core_number
        )
```

Notebook

Input ∧

+ Add Input   ⬆ Upload

DATASETS

▸ 🟦 snlp-project-sst2

Output ∧

▸ 📁 /kaggle/working

Session options ∧

ACCELERATOR
GPU T4 ×2 ▾

Quota: 25:03 / 30 hrs

LANGUAGE
Python ▾

PERSISTENCE
Files only ▾

Training on GPU has a quota of 30 hours every week, and it is reset at every Saturday.
You cannot run GPU when you run out of quota, and you must until next Saturday for it to reset

Task 2_DeBERTa_SST2   Draft saved

File   Edit   View   Run   Add-ons   Help

👥 Share   🕒 Save Version   2

+   ✂   📋   📄   ▶   ▶▶   Run All   Code ▾   ● Draft Session (3m)   H D D | C P U | R A M | G P U | G P U   ⏻   🔄   ⋮

Notebook

❓ 5 hours remaining on your GPU T4 ×2 quota this week. Turn off GPU T4 ×2 to prevent exceeding this quota.   ✕

In reality, it is not required to run this cell. Obtaining the results is already enough. The saved model would be extremely heavy, and there are already fine tuned BERT models on HuggingFace

```python
from pathlib import Path

output_dir = Path("__file__").parents[0].absolute().joinpath(f"{bert_model_name}/model_save")
output_dir.mkdir(exist_ok=True)

# Save a trained model, configuration and tokenizer using `save_pretrained()`.
# They can then be reloaded using `from_pretrained()`

# model_to_save = bert_classification_model.module if hasattr(bert_classification_model, 'module')
# model_to_save.save_pretrained(output_dir)
# tokenizer.save_pretrained(str(output_dir.absolute()))
```

+ Code   + Markdown

Input   ⌃

+ Add Input   ⭡ Upload

DATASETS

▸ 🔷 snlp-project-sst2

Output (56KB / 19.5GB)   ⌃

▸ 📁 /kaggle/working   ↻

Session options   ⌃

ACCELERATOR

GPU T4 ×2   ▾

Quota: 25:08 / 30 hrs

LANGUAGE

Python   ▾

PERSISTENCE

Files only   ▾