t81_558_class_11_01_huggingface

June 23, 2025

1

2 T81-558: Applications of Deep Neural Networks

Module 11: Natural Language Processing with Hugging Face * Instructor: Jeff Heaton, McKelvey School of Engineering, Washington University in St. Louis * For more information visit the class website.

3 Module 11 Material

- **Part 11.1: Introduction to Hugging Face** [Video] [Notebook]
- Part 11.2: Hugging Face Tokenizers [Video] [Notebook]
- Part 11.3: Hugging Face Datasets [Video] [Notebook]
- Part 11.4: Training Hugging Face Models [Video] [Notebook]
- Part 11.5: What are Embedding Layers in Keras [Video] [Notebook]

4 Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

[1]:
```
try:
    %tensorflow_version 2.x
    COLAB = True
    print("Note: using Google CoLab")
except:
    print("Note: not using Google CoLab")
    COLAB = False
```

```
Note: not using Google CoLab
```

5 Part 11.1: Introduction to Hugging Face

Transformers have become a mainstay of natural language processing. This module will examine the Hugging Face Python library for natural language processing, bringing together pretrained transformers, data sets, tokenizers, and other elements. Through the Hugging Face API, you can quickly begin using sentiment analysis, entity recognition, language translation, summarization, and text generation.

Colab does not install Hugging face by default. Whether installing Hugging Face directly into a local computer or utilizing it through Colab, the following commands will install the library.

[2]:
```
# HIDE OUTPUT
import tensorflow as tf
print(f"Tensor Flow Version: {tf.__version__}")

!pip install transformers
!pip install transformers[sentencepiece]
```

```
2024-02-14 20:42:53.807584: I tensorflow/core/util/port.cc:113] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-02-14 20:42:53.847346: E
external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register
cuDNN factory: Attempting to register factory for plugin cuDNN when one has
already been registered
2024-02-14 20:42:53.847373: E
external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
2024-02-14 20:42:53.848137: E
external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to
register cuBLAS factory: Attempting to register factory for plugin cuBLAS when
one has already been registered
2024-02-14 20:42:53.853359: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other
operations, rebuild TensorFlow with the appropriate compiler flags.
2024-02-14 20:42:54.663206: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT

Tensor Flow Version: 2.15.0
Defaulting to user installation because normal site-packages is not writeable
Collecting transformers
  Downloading transformers-4.37.2-py3-none-any.whl (8.4 MB)
                             8.4/8.4 MB
3.7 MB/s eta 0:00:0000:0100:01
Collecting tokenizers<0.19,>=0.14
  Downloading
tokenizers-0.15.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(3.6 MB)
                             3.6/3.6 MB
5.6 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: packaging>=20.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (23.2)
```

```
Collecting regex!=2019.12.17
  Downloading
regex-2023.12.25-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (773
kB)
                           774.0/774.0
KB 8.4 MB/s eta 0:00:00a 0:00:01
Requirement already satisfied: pyyaml>=5.1 in /usr/lib/python3/dist-
packages (from transformers) (5.4.1)
Collecting tqdm>=4.27
  Downloading tqdm-4.66.2-py3-none-any.whl (78 kB)
                           78.3/78.3 KB
5.8 MB/s eta 0:00:00
Requirement already satisfied: filelock in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (3.12.4)
Collecting huggingface-hub<1.0,>=0.19.3
  Downloading huggingface_hub-0.20.3-py3-none-any.whl (330 kB)
                           330.1/330.1
KB 9.0 MB/s eta 0:00:0000:01
Requirement already satisfied: requests in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (2.31.0)
Collecting safetensors>=0.4.1
  Downloading
safetensors-0.4.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.3 MB)
                           1.3/1.3 MB
9.4 MB/s eta 0:00:00ta 0:00:01
Requirement already satisfied: numpy>=1.17 in
/home/tmeng12/.local/lib/python3.10/site-packages (from transformers) (1.26.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers) (4.8.0)
Requirement already satisfied: fsspec>=2023.5.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers) (2023.9.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests->transformers) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests->transformers) (2020.6.20)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-
packages (from requests->transformers) (1.26.5)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/tmeng12/.local/lib/python3.10/site-packages (from requests->transformers)
(3.3.0)
Installing collected packages: tqdm, safetensors, regex, huggingface-hub,
tokenizers, transformers
Successfully installed huggingface-hub-0.20.3 regex-2023.12.25 safetensors-0.4.2
```

```
tokenizers-0.15.2 tqdm-4.66.2 transformers-4.37.2
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: transformers[sentencepiece] in
/home/tmeng12/.local/lib/python3.10/site-packages (4.37.2)
Requirement already satisfied: safetensors>=0.4.1 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (0.4.2)
Requirement already satisfied: packaging>=20.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (23.2)
Requirement already satisfied: tqdm>=4.27 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (4.66.2)
Requirement already satisfied: tokenizers<0.19,>=0.14 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (0.15.2)
Requirement already satisfied: numpy>=1.17 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (1.26.3)
Requirement already satisfied: pyyaml>=5.1 in /usr/lib/python3/dist-packages
(from transformers[sentencepiece]) (5.4.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (0.20.3)
Requirement already satisfied: requests in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (2.31.0)
Requirement already satisfied: regex!=2019.12.17 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (2023.12.25)
Requirement already satisfied: filelock in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (3.12.4)
Requirement already satisfied: protobuf in
/home/tmeng12/.local/lib/python3.10/site-packages (from
transformers[sentencepiece]) (4.23.4)
Collecting sentencepiece!=0.1.92,>=0.1.91
  Downloading
sentencepiece-0.1.99-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.3 MB)
                           1.3/1.3 MB
3.0 MB/s eta 0:00:0000:0100:010m
Requirement already satisfied: fsspec>=2023.5.0 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers[sentencepiece]) (2023.9.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/tmeng12/.local/lib/python3.10/site-packages (from huggingface-
hub<1.0,>=0.19.3->transformers[sentencepiece]) (4.8.0)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/tmeng12/.local/lib/python3.10/site-packages (from
requests->transformers[sentencepiece]) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/lib/python3/dist-packages
(from requests->transformers[sentencepiece]) (3.3)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/lib/python3/dist-
packages (from requests->transformers[sentencepiece]) (1.26.5)
Requirement already satisfied: certifi>=2017.4.17 in /usr/lib/python3/dist-
packages (from requests->transformers[sentencepiece]) (2020.6.20)
Installing collected packages: sentencepiece
Successfully installed sentencepiece-0.1.99
```

Now that we have Hugging Face installed, the following sections will demonstrate how to apply Hugging Face to a variety of everyday tasks. After this introduction, the remainder of this module will take a deeper look at several specific NLP tasks applied to Hugging Face.

5.1 Sentiment Analysis

Sentiment analysis uses natural language processing, text analysis, computational linguistics, and biometrics to identify the tone of written text. Passages of written text can be into simple binary states of positive or negative tone. More advanced sentiment analysis might classify text into additional categories: sadness, joy, love, anger, fear, or surprise.

To demonstrate sentiment analysis, we begin by loading sample text, Shakespeare's 18th sonnet, a famous poem.

[3]:
```
from urllib.request import urlopen

# Read sample text, a poem
URL = "https://data.heatonresearch.com/data/t81-558/"\
    "datasets/sonnet_18.txt"
f = urlopen(URL)
text = f.read().decode("utf-8")
print(text)
```

```
Sonnet 18 original text
William Shakespeare

Shall I compare thee to a summer's day?
Thou art more lovely and more temperate:
Rough winds do shake the darling buds of May,
And summer's lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd;
And every fair from fair sometime declines,
By chance or nature's changing course untrimm'd;
But thy eternal summer shall not fade
Nor lose possession of that fair thou owest;
Nor shall Death brag thou wander'st in his shade,
```

```
When in eternal lines to time thou growest:
So long as men can breathe or eyes can see,
So long lives this and this gives life to thee.
```

Usually, you have to preprocess text into embeddings or other vector forms before presentation to a neural network. Hugging Face provides a pipeline that simplifies this process greatly. The pipeline allows you to pass regular Python strings to the transformers and return standard Python values.

We begin by loading a text-classification model. We do not specify the exact model type wanted, so Hugging Face automatically chooses a network from the Hugging Face hub named:

- distilbert-base-uncased-finetuned-sst-2-english

To specify the model to use, pass the model parameter, such as:

```
pipe = pipeline(model="roberta-large-mnli")
```

The following code loads a model pipeline and a model for sentiment analysis.

[4]:
```
# HIDE OUTPUT
import pandas as pd
from transformers import pipeline

classifier = pipeline("text-classification")
```

```
No model was supplied, defaulted to distilbert-base-uncased-finetuned-
sst-2-english and revision af0f99b (https://huggingface.co/distilbert-base-
uncased-finetuned-sst-2-english).
Using a pipeline without specifying a model name and revision in production is
not recommended.
```

We can now display the sentiment analysis results with a Pandas dataframe.

[ ]:
```
outputs = classifier(text)
df=pd.DataFrame(outputs)
df
```

[ ]:
```
      label     score
0  POSITIVE  0.984666
```

As you can see, the poem was considered 0.98 positive.

5.2 Entity Tagging

Entity tagging is the process that takes source text and finds parts of that text that represent entities, such as one of the following:

- Location (LOC)
- Organizations (ORG)
- Person (PER)
- Miscellaneous (MISC)

The following code requests a "named entity recognizer" (ner) and processes the specified text.

[6]:
```
# HIDE OUTPUT
text2 = "Abraham Lincoln was a president who lived in the United States."

tagger = pipeline("ner", aggregation_strategy="simple")
```

```
No model was supplied, defaulted to dbmdz/bert-large-cased-finetuned-
conll03-english and revision f2482bf (https://huggingface.co/dbmdz/bert-large-
cased-finetuned-conll03-english).
Using a pipeline without specifying a model name and revision in production is
not recommended.

model.safetensors:   0%|          | 0.00/1.33G [00:00<?, ?B/s]
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
/media/data/github/learn/ai/jheaton/t81_558_deep_learning/
 ↪t81_558_class_11_01_huggingface.ipynb Cell 15 line 4

     <a href='vscode-notebook-cell:/media/data/github/learn/ai/jheaton/
 ↪t81_558_deep_learning/t81_558_class_11_01_huggingface.ipynb#X20sZmlsZQ%3D%3D?
 ↪line=0'>1</a> # HIDE OUTPUT
     <a href='vscode-notebook-cell:/media/data/github/learn/ai/jheaton/
 ↪t81_558_deep_learning/t81_558_class_11_01_huggingface.ipynb#X20sZmlsZQ%3D%3D?
 ↪line=1'>2</a> text2 = "Abraham Lincoln was a president who lived in the Unite␣
 ↪States."
----> <a href='vscode-notebook-cell:/media/data/github/learn/ai/jheaton/
 ↪t81_558_deep_learning/t81_558_class_11_01_huggingface.ipynb#X20sZmlsZQ%3D%3D?
 ↪line=3'>4</a> tagger = pipeline("ner", aggregation_strategy="simple")

File ~/.local/lib/python3.10/site-packages/transformers/pipelines/__init__.py:
 ↪870, in pipeline(task, model, config, tokenizer, feature_extractor,␣
 ↪image_processor, framework, revision, use_fast, token, device, device_map,␣
 ↪torch_dtype, trust_remote_code, model_kwargs, pipeline_class, **kwargs)
    868 if isinstance(model, str) or framework is None:
    869     model_classes = {"tf": targeted_task["tf"], "pt":␣
 ↪targeted_task["pt"]}
--> 870     framework, model = infer_framework_load_model(
    871         model,
    872         model_classes=model_classes,
    873         config=config,
    874         framework=framework,
    875         task=task,
    876         **hub_kwargs,
    877         **model_kwargs,
    878     )
    880 model_config = model.config
    881 hub_kwargs["_commit_hash"] = model.config._commit_hash

File ~/.local/lib/python3.10/site-packages/transformers/pipelines/base.py:278,␣
 ↪in infer_framework_load_model(model, config, model_classes, task, framework,␣
 ↪**model_kwargs)
```

```
    272     logger.warning(
    273         "Model might be a PyTorch model (ending with `.bin`) but PyTorch␣
 ↪is not available. "
    274         "Trying to load the model with Tensorflow."
    275     )
    277 try:
--> 278     model = model_class.from_pretrained(model, **kwargs)
    279     if hasattr(model, "eval"):
    280         model = model.eval()

File ~/.local/lib/python3.10/site-packages/transformers/models/auto/auto_factory.
 ↪py:566, in _BaseAutoModelClass.from_pretrained(cls,␣
 ↪pretrained_model_name_or_path, *model_args, **kwargs)
    564 elif type(config) in cls._model_mapping.keys():
    565     model_class = _get_model_class(config, cls._model_mapping)
--> 566     return model_class.from_pretrained(
    567         pretrained_model_name_or_path, *model_args, config=config,␣
 ↪**hub_kwargs, **kwargs
    568     )
    569 raise ValueError(
    570     f"Unrecognized configuration class {config.__class__} for this kind␣
 ↪of AutoModel: {cls.__name__}.\n"
    571     f"Model type should be one of {', '.join(c.__name__ for c in cls.
 ↪_model_mapping.keys())}."
    572 )

File ~/.local/lib/python3.10/site-packages/transformers/modeling_utils.py:3383,␣
 ↪in PreTrainedModel.from_pretrained(cls, pretrained_model_name_or_path, config␣
 ↪cache_dir, ignore_mismatched_sizes, force_download, local_files_only, token,␣
 ↪revision, use_safetensors, *model_args, **kwargs)
   3368 try:
   3369     # Load from URL or cache if already cached
   3370     cached_file_kwargs = {
   3371         "cache_dir": cache_dir,
   3372         "force_download": force_download,
   (…)
   3381         "_commit_hash": commit_hash,
   3382     }
-> 3383     resolved_archive_file = cached_file(pretrained_model_name_or_path,␣
 ↪filename, **cached_file_kwargs)
   3385     # Since we set _raise_exceptions_for_missing_entries=False, we don't␣
 ↪get an exception but a None
   3386     # result when internet is up, the repo and revision exist, but the␣
 ↪file does not.
   3387     if resolved_archive_file is None and filename ==␣
 ↪_add_variant(SAFE_WEIGHTS_NAME, variant):
   3388         # Maybe the checkpoint is sharded, we try to grab the index name␣
 ↪in this case.
```

```
File ~/.local/lib/python3.10/site-packages/transformers/utils/hub.py:385, in␣
 ↪cached_file(path_or_repo_id, filename, cache_dir, force_download,␣
 ↪resume_download, proxies, token, revision, local_files_only, subfolder,␣
 ↪repo_type, user_agent, _raise_exceptions_for_missing_entries,␣
 ↪_raise_exceptions_for_connection_errors, _commit_hash, **deprecated_kwargs)
    382 user_agent = http_user_agent(user_agent)
    383 try:
    384     # Load from URL or cache if already cached
--> 385     resolved_file = hf_hub_download(
    386         path_or_repo_id,
    387         filename,
    388         subfolder=None if len(subfolder) == 0 else subfolder,
    389         repo_type=repo_type,
    390         revision=revision,
    391         cache_dir=cache_dir,
    392         user_agent=user_agent,
    393         force_download=force_download,
    394         proxies=proxies,
    395         resume_download=resume_download,
    396         token=token,
    397         local_files_only=local_files_only,
    398     )
    399 except GatedRepoError as e:
    400     raise EnvironmentError(
    401         "You are trying to access a gated repo.\nMake sure to request␣
 ↪access at "
    402         f"https://huggingface.co/{path_or_repo_id} and pass a token␣
 ↪having permission to this repo either "
    403         "by logging in with `huggingface-cli login` or by passing␣
 ↪`token=<your_token>`."
    404     ) from e

File ~/.local/lib/python3.10/site-packages/huggingface_hub/utils/_validators.py
 ↪118, in validate_hf_hub_args.<locals>._inner_fn(*args, **kwargs)
    115 if check_use_auth_token:
    116     kwargs = smoothly_deprecate_use_auth_token(fn_name=fn.__name__,␣
 ↪has_token=has_token, kwargs=kwargs)
--> 118 return fn(*args, **kwargs)

File ~/.local/lib/python3.10/site-packages/huggingface_hub/file_download.py:
 ↪1457, in hf_hub_download(repo_id, filename, subfolder, repo_type, revision,␣
 ↪library_name, library_version, cache_dir, local_dir, local_dir_use_symlinks,␣
 ↪user_agent, force_download, force_filename, proxies, etag_timeout,␣
 ↪resume_download, token, local_files_only, legacy_cache_layout, endpoint)
   1454         if local_dir is not None:
   1455             _check_disk_space(expected_size, local_dir)
-> 1457     http_get(
   1458         url_to_download,
```

```
   1459         temp_file,
   1460         proxies=proxies,
   1461         resume_size=resume_size,
   1462         headers=headers,
   1463         expected_size=expected_size,
   1464     )
   1466 if local_dir is None:
   1467     logger.debug(f"Storing {url} in cache at {blob_path}")

File ~/.local/lib/python3.10/site-packages/huggingface_hub/file_download.py:524
 ↪in http_get(url, temp_file, proxies, resume_size, headers, expected_size,␣
 ↪_nb_retries)
    522 new_resume_size = resume_size
    523 try:
--> 524     for chunk in r.iter_content(chunk_size=DOWNLOAD_CHUNK_SIZE):
    525         if chunk:  # filter out keep-alive new chunks
    526             progress.update(len(chunk))

File ~/.local/lib/python3.10/site-packages/requests/models.py:816, in Response.
 ↪iter_content.<locals>.generate()
    814 if hasattr(self.raw, "stream"):
    815     try:
--> 816         yield from self.raw.stream(chunk_size, decode_content=True)
    817     except ProtocolError as e:
    818         raise ChunkedEncodingError(e)

File /usr/lib/python3/dist-packages/urllib3/response.py:576, in HTTPResponse.
 ↪stream(self, amt, decode_content)
    574 else:
    575     while not is_fp_closed(self._fp):
--> 576         data = self.read(amt=amt, decode_content=decode_content)
    578         if data:
    579             yield data

File /usr/lib/python3/dist-packages/urllib3/response.py:519, in HTTPResponse.
 ↪read(self, amt, decode_content, cache_content)
    517 else:
    518     cache_content = False
--> 519     data = self._fp.read(amt) if not fp_closed else b""
    520     if (
    521         amt != 0 and not data
    522     ):  # Platform-specific: Buggy versions of Python.
   (…)
    528         # not properly close the connection in all cases. There is
    529         # no harm in redundantly calling close.
    530         self._fp.close()

File /usr/lib/python3.10/http/client.py:466, in HTTPResponse.read(self, amt)
```

```
    463 if self.length is not None and amt > self.length:
    464     # clip the read to the "end of response"
    465     amt = self.length
--> 466 s = self.fp.read(amt)
    467 if not s and amt:
    468     # Ideally, we would raise IncompleteRead if the content-length
    469     # wasn't satisfied, but it might break compatibility.
    470     self._close_conn()

File /usr/lib/python3.10/socket.py:705, in SocketIO.readinto(self, b)
    703 while True:
    704     try:
--> 705         return self._sock.recv_into(b)
    706     except timeout:
    707         self._timeout_occurred = True

File /usr/lib/python3.10/ssl.py:1303, in SSLSocket.recv_into(self, buffer,␣
 ↪nbytes, flags)
   1299     if flags != 0:
   1300         raise ValueError(
   1301           "non-zero flags not allowed in calls to recv_into() on %s" %
   1302           self.__class__)
-> 1303     return self.read(nbytes, buffer)
   1304 else:
   1305     return super().recv_into(buffer, nbytes, flags)

File /usr/lib/python3.10/ssl.py:1159, in SSLSocket.read(self, len, buffer)
   1157 try:
   1158     if buffer is not None:
-> 1159         return self._sslobj.read(len, buffer)
   1160     else:
   1161         return self._sslobj.read(len)

KeyboardInterrupt:
```

We similarly view the results as a Pandas data frame. As you can see, the person (PER) of Abraham Lincoln and location (LOC) of the United States is recognized.

[ ]:
```
outputs = tagger(text2)
pd.DataFrame(outputs)
```

5.3 Question Answering

Another common task for NLP is question answering from a reference text. We load such a model with the following code.

```
[ ]: # HIDE OUTPUT
reader = pipeline("question-answering")
question = "What now shall fade?"
```

For this example, we will pose the question "what shall fade" to Hugging Face for Sonnet 18. We see the correct answer of "eternal summer."

```
[ ]: outputs = reader(question=question, context=text)
pd.DataFrame([outputs])
```

5.4 Language Translation

Language translation is yet another common task for NLP and Hugging Face.

```
[ ]: # HIDE OUTPUT
translator = pipeline("translation_en_to_de",
                      model="Helsinki-NLP/opus-mt-en-de")
```

The following code translates Sonnet 18 from English into German.

```
[ ]: outputs = translator(text, clean_up_tokenization_spaces=True,
                         min_length=100)
print(outputs[0]['translation_text'])
```

5.5 Summarization

Summarization is an NLP task that summarizes a more lengthy text into just a few sentences.

```
[ ]: # HIDE OUTPUT
text2 = """
An apple is an edible fruit produced by an apple tree (Malus domestica).
Apple trees are cultivated worldwide and are the most widely grown species
in the genus Malus. The tree originated in Central Asia, where its wild
ancestor, Malus sieversii, is still found today. Apples have been grown
for thousands of years in Asia and Europe and were brought to North America
by European colonists. Apples have religious and mythological significance
in many cultures, including Norse, Greek, and European Christian tradition.
"""

summarizer = pipeline("summarization")
```

The following code summarizes the Wikipedia entry for an "apple."

```
[ ]: outputs = summarizer(text2, max_length=45,
                          clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

5.6 Text Generation

Finally, text generation allows us to take an input text and request the pretrained neural network to continue that text.

[ ]:
```
# HIDE OUTPUT
from urllib.request import urlopen

generator = pipeline("text-generation")
```

Here an example is provided that generates additional text after Sonnet 18.

[ ]:
```
outputs = generator(text, max_length=400)
print(outputs[0]['generated_text'])
```

Open in Colab

T81-558: Applications of Deep Neural Networks

Module 11: Natural Language Processing with Hugging Face

- Instructor: Jeff Heaton, McKelvey School of Engineering, Washington University in St. Louis
- For more information visit the class website.

Module 11 Material

- Part 11.1: Introduction to Hugging Face [Video] [Notebook]
- **Part 11.2: Hugging Face Tokenizers** [Video] [Notebook]
- Part 11.3: Hugging Face Datasets [Video] [Notebook]
- Part 11.4: Training Hugging Face Models [Video] [Notebook]
- Part 11.5: What are Embedding Layers in Keras [Video] [Notebook]

Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

In [ ]:
```
try:
    %tensorflow_version 2.x
    COLAB = True
    print("Note: using Google CoLab")
except:
    print("Note: not using Google CoLab")
    COLAB = False
```

```
Note: using Google CoLab
```

Part 11.2: Hugging Face Tokenizers

Tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Consider how the program might break up the following sentences into words.

- This is a test.

- Ok, but what about this?
- Is U.S.A. the same as USA.?
- What is the best data-set to use?
- I think I will do this-no wait; I will do that.

The hugging face includes tokenizers that can break these sentences into words and subwords. Because English, and some other languages, are made up of common word parts, we tokenize subwords. For example, a gerund word, such as "sleeping," will be tokenized into "sleep" and "##ing".

We begin by installing Hugging Face if needed.

In [ ]:
```
# HIDE OUTPUT
!pip install transformers
!pip install transformers[sentencepiece]
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.7/dist
-packages (4.17.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-pac
kages (from transformers) (2.23.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-pac
kages (from transformers) (3.6.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.
7/dist-packages (from transformers) (2019.12.20)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/dist-
packages (from transformers) (6.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/d
ist-packages (from transformers) (21.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from transformers) (1.21.5)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-p
ackages (from transformers) (4.63.0)
Requirement already satisfied: tokenizers!=0.11.3,>=0.11.1 in /usr/local/li
b/python3.7/dist-packages (from transformers) (0.11.6)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in /usr/local/li
b/python3.7/dist-packages (from transformers) (0.4.0)
Requirement already satisfied: sacremoses in /usr/local/lib/python3.7/dist-p
ackages (from transformers) (0.0.49)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.
7/dist-packages (from transformers) (4.11.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/
python3.7/dist-packages (from huggingface-hub<1.0,>=0.1.0->transformers) (3.
10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/py
thon3.7/dist-packages (from packaging>=20.0->transformers) (3.0.7)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-pa
ckages (from importlib-metadata->transformers) (3.7.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.
7/dist-packages (from requests->transformers) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.
7/dist-packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist
-packages (from requests->transformers) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /u
sr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packag
es (from sacremoses->transformers) (7.1.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.15.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packa
ges (from sacremoses->transformers) (1.1.0)
Requirement already satisfied: transformers[sentencepiece] in /usr/local/li
b/python3.7/dist-packages (4.17.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-p
ackages (from transformers[sentencepiece]) (4.63.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-pac
kages (from transformers[sentencepiece]) (2.23.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in /usr/local/li
b/python3.7/dist-packages (from transformers[sentencepiece]) (0.4.0)
Requirement already satisfied: sacremoses in /usr/local/lib/python3.7/dist-p
ackages (from transformers[sentencepiece]) (0.0.49)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
```

```
packages (from transformers[sentencepiece]) (1.21.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/d
ist-packages (from transformers[sentencepiece]) (21.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.
7/dist-packages (from transformers[sentencepiece]) (4.11.3)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (6.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-pac
kages (from transformers[sentencepiece]) (3.6.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.
7/dist-packages (from transformers[sentencepiece]) (2019.12.20)
Requirement already satisfied: tokenizers!=0.11.3,>=0.11.1 in /usr/local/li
b/python3.7/dist-packages (from transformers[sentencepiece]) (0.11.6)
Requirement already satisfied: sentencepiece!=0.1.92,>=0.1.91 in /usr/local/
lib/python3.7/dist-packages (from transformers[sentencepiece]) (0.1.96)
Requirement already satisfied: protobuf in /usr/local/lib/python3.7/dist-pac
kages (from transformers[sentencepiece]) (3.17.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/
python3.7/dist-packages (from huggingface-hub<1.0,>=0.1.0->transformers[sent
encepiece]) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/py
thon3.7/dist-packages (from packaging>=20.0->transformers[sentencepiece])
(3.0.7)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-pa
ckages (from importlib-metadata->transformers[sentencepiece]) (3.7.0)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.7/dist-pac
kages (from protobuf->transformers[sentencepiece]) (1.15.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist
-packages (from requests->transformers[sentencepiece]) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.
7/dist-packages (from requests->transformers[sentencepiece]) (2021.10.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /u
sr/local/lib/python3.7/dist-packages (from requests->transformers[sentencepi
ece]) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.
7/dist-packages (from requests->transformers[sentencepiece]) (3.0.4)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packa
ges (from sacremoses->transformers[sentencepiece]) (1.1.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packag
es (from sacremoses->transformers[sentencepiece]) (7.1.2)
```

First, we create a Hugging Face tokenizer. There are several different tokenizers available from the Hugging Face hub. For this example, we will make use of the following tokenizer:

- distilbert-base-uncased

This tokenizer is based on BERT and assumes case-insensitive English text.

In [ ]:
```
from transformers import AutoTokenizer
model = "distilbert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model)
```

We can now tokenize a sample sentence.

In [ ]:
```
encoded = tokenizer('Tokenizing text is easy.')
print(encoded)
```

```
{'input_ids': [101, 19204, 6026, 3793, 2003, 3733, 1012, 102], 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1]}
```

The result of this tokenization contains two elements:

- input_ids - The individual subword indexes, each index uniquely identifies a subword.
- attention_mask - Which values in *input_ids* are meaningful and not padding. This sentence had no padding, so all elements have an attention mask of "1". Later, we will request the output to be of a fixed length, introducing padding, which always has an attention mask of "0". Though each tokenizer can be implemented differently, the attention mask of a tokenizer is generally either "0" or "1".

Due to subwords and special tokens, the number of tokens may not match the number of words in the source string. We can see the meanings of the individual tokens by converting these IDs back to strings.

In [ ]:
```
tokenizer.convert_ids_to_tokens(encoded.input_ids)
```

Out[ ]:
```
['[CLS]', 'token', '##izing', 'text', 'is', 'easy', '.', '[SEP]']
```

As you can see, there are two special tokens placed at the beginning and end of each sequence. We will soon see how we can include or exclude these special tokens. These special tokens can vary per tokenizer; however, [CLS] begins a sequence for this tokenizer, and [SEP] ends a sequence. You will also see that the gerund "tokening" is broken into "token" and "*ing".

For this tokenizer, the special tokens occur between 100 and 103. Most Hugging Face tokenizers use this approximate range for special tokens. The value zero (0) typically represents padding. We can display all special tokens with this command.

In [ ]:
```
tokenizer.convert_ids_to_tokens([0, 100, 101, 102, 103])
```

Out[ ]:
```
['[PAD]', '[UNK]', '[CLS]', '[SEP]', '[MASK]']
```

This tokenizer supports these common tokens:

- [CLS] - Sequence beginning.
- [SEP] - Sequence end.
- [PAD] - Padding.
- [UNK] - Unknown token.
- [MASK] - Mask out tokens for a neural network to predict. Not used in this book, see MLM paper.

It is also possible to tokenize lists of sequences. We can pad and truncate sequences to achieve a standard length by tokenizing many sequences at once.

In [ ]:
```
text = [
    "This movie was great!",
    "I hated this move, waste of time!",
    "Epic?"
]

encoded = tokenizer(text, padding=True, add_special_tokens=True)

print("**Input IDs**")
for a in encoded.input_ids:
    print(a)

print("**Attention Mask**")
for a in encoded.attention_mask:
    print(a)
```

```
**Input IDs**
[101, 2023, 3185, 2001, 2307, 999, 102, 0, 0, 0, 0]
[101, 1045, 6283, 2023, 2693, 1010, 5949, 1997, 2051, 999, 102]
[101, 8680, 1029, 102, 0, 0, 0, 0, 0, 0, 0]
**Attention Mask**
[1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0]
```

Notice the **input_id**'s for the three movie review text sequences. Each of these sequences begins with 101 and we pad with zeros. Just before the padding, each group of IDs ends with 102. The attention masks also have zeros for each of the padding entries.

We used two parameters to the tokenizer to control the tokenization process. Some other useful parameters include:

- add_special_tokens (defaults to True) Whether or not to encode the sequences with the special tokens relative to their model.
- padding (defaults to False) Activates and controls truncation.
- max_length (optional) Controls the maximum length to use by one of the truncation/padding parameters.

In [ ]:
```
```

t81_558_class_11_03_hf_datasets

June 23, 2025

1 T81-558: Applications of Deep Neural Networks

Module 11: Natural Language Processing with Hugging Face * Instructor: Jeff Heaton, McKelvey School of Engineering, Washington University in St. Louis * For more information visit the class website.

2 Module 11 Material

- Part 11.1: Introduction to Hugging Face [Video] [Notebook]
- Part 11.2: Hugging Face Tokenizers [Video] [Notebook]
- **Part 11.3: Hugging Face Datasets** [Video] [Notebook]
- Part 11.4: Training Hugging Face Models [Video] [Notebook]
- Part 11.5: What are Embedding Layers in Keras [Video] [Notebook]

3 Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

[ ]:
```
try:
    %tensorflow_version 2.x
    COLAB = True
    print("Note: using Google CoLab")
except:
    print("Note: not using Google CoLab")
    COLAB = False
```

```
Note: using Google CoLab
```

4 Part 11.3: Hugging Face Datasets

The Hugging Face hub includes data sets useful for natural language processing (NLP). The Hugging Face library provides functions that allow you to navigate and obtain these data sets. When we access Hugging Face data sets, the data is in a format specific to Hugging Face. In this part, we will explore this format and see how to convert it to Pandas or TensorFlow data.

We begin by installing Hugging Face if needed. It is also essential to install Hugging Face datasets.

[ ]:
```
# HIDE OUTPUT
!pip install transformers
!pip install transformers[sentencepiece]
!pip install datasets
```

```
Collecting transformers
  Downloading transformers-4.17.0-py3-none-any.whl (3.8 MB)
     |                  | 3.8 MB 5.1 MB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from transformers) (3.6.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from transformers) (1.21.5)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.7/dist-packages (from transformers) (2019.12.20)
Collecting tokenizers!=0.11.3,>=0.11.1
  Downloading
tokenizers-0.11.6-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (6.5
MB)
     |                  | 6.5 MB 55.4 MB/s
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.4.0-py3-none-any.whl (67 kB)
     |                  | 67 kB 7.3 MB/s
Collecting sacremoses
  Downloading sacremoses-0.0.49-py3-none-any.whl (895 kB)
     |                  | 895 kB 55.9 MB/s
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from transformers) (4.11.3)
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (596 kB)
     |                  | 596 kB 61.0 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from transformers) (2.23.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-
packages (from transformers) (4.63.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-
packages (from transformers) (21.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0,>=0.1.0->transformers) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging>=20.0->transformers)
(3.0.7)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->transformers) (3.7.0)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (3.0.4)
```

```
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->transformers) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.1.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (7.1.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.15.0)
Installing collected packages: pyyaml, tokenizers, sacremoses, huggingface-hub,
transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.4.0 pyyaml-6.0 sacremoses-0.0.49
tokenizers-0.11.6 transformers-4.17.0
Requirement already satisfied: transformers[sentencepiece] in
/usr/local/lib/python3.7/dist-packages (4.17.0)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (2.23.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (21.3)
Requirement already satisfied: sacremoses in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (0.0.49)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(0.4.0)
Requirement already satisfied: tokenizers!=0.11.3,>=0.11.1 in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(0.11.6)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (4.63.0)
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(4.11.3)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(2019.12.20)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (6.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (3.6.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (1.21.5)
```

```
Collecting sentencepiece!=0.1.92,>=0.1.91
  Downloading
sentencepiece-0.1.96-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.2 MB)
     |                  | 1.2 MB 5.1 MB/s
Requirement already satisfied: protobuf in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (3.17.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0,>=0.1.0->transformers[sentencepiece]) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from
packaging>=20.0->transformers[sentencepiece]) (3.0.7)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->transformers[sentencepiece]) (3.7.0)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.7/dist-
packages (from protobuf->transformers[sentencepiece]) (1.15.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->transformers[sentencepiece]) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from
requests->transformers[sentencepiece]) (2021.10.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from
requests->transformers[sentencepiece]) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from
requests->transformers[sentencepiece]) (3.0.4)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers[sentencepiece]) (1.1.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers[sentencepiece]) (7.1.2)
Installing collected packages: sentencepiece
Successfully installed sentencepiece-0.1.96
Collecting datasets
  Downloading datasets-2.0.0-py3-none-any.whl (325 kB)
     |                  | 325 kB 5.0 MB/s
Collecting xxhash
  Downloading
xxhash-3.0.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (212 kB)
     |                  | 212 kB 86.4 MB/s
Requirement already satisfied: dill in /usr/local/lib/python3.7/dist-
packages (from datasets) (0.3.4)
Requirement already satisfied: huggingface-hub<1.0.0,>=0.1.0 in
/usr/local/lib/python3.7/dist-packages (from datasets) (0.4.0)
Requirement already satisfied: pyarrow>=5.0.0 in /usr/local/lib/python3.7/dist-
packages (from datasets) (6.0.1)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.7/dist-
```

```
packages (from datasets) (4.63.0)
Collecting responses<0.19
  Downloading responses-0.18.0-py3-none-any.whl (38 kB)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-
packages (from datasets) (21.3)
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from datasets) (4.11.3)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from datasets) (1.21.5)
Requirement already satisfied: multiprocess in /usr/local/lib/python3.7/dist-
packages (from datasets) (0.70.12.2)
Requirement already satisfied: requests>=2.19.0 in
/usr/local/lib/python3.7/dist-packages (from datasets) (2.23.0)
Collecting aiohttp
  Downloading aiohttp-3.8.1-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (1.1 MB)
     |                  | 1.1 MB 70.0 MB/s
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-
packages (from datasets) (1.3.5)
Collecting fsspec[http]>=2021.05.0
  Downloading fsspec-2022.2.0-py3-none-any.whl (134 kB)
     |                  | 134 kB 87.8 MB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from huggingface-hub<1.0.0,>=0.1.0->datasets) (3.6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0.0,>=0.1.0->datasets) (3.10.0.2)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages
(from huggingface-hub<1.0.0,>=0.1.0->datasets) (6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging->datasets) (3.0.7)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.19.0->datasets) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests>=2.19.0->datasets) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.19.0->datasets)
(1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.19.0->datasets)
(2021.10.8)
Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
  Downloading urllib3-1.25.11-py2.py3-none-any.whl (127 kB)
     |                  | 127 kB 89.0 MB/s
Collecting multidict<7.0,>=4.5
  Downloading
multidict-6.0.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (94
```

```
kB)
     |                  | 94 kB 4.7 MB/s
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Collecting asynctest==0.13.0
  Downloading asynctest-0.13.0-py3-none-any.whl (26 kB)
Collecting frozenlist>=1.1.1
  Downloading frozenlist-1.3.0-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (144 kB)
     |                  | 144 kB 88.6 MB/s
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.2.0-py3-none-any.whl (8.2 kB)
Collecting yarl<2.0,>=1.0
  Downloading yarl-1.7.2-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (271 kB)
     |                  | 271 kB 87.1 MB/s
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.7/dist-packages (from aiohttp->datasets) (21.4.0)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in
/usr/local/lib/python3.7/dist-packages (from aiohttp->datasets) (2.0.12)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->datasets) (3.7.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
packages (from pandas->datasets) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.7/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-
packages (from python-dateutil>=2.7.3->pandas->datasets) (1.15.0)
Installing collected packages: multidict, frozenlist, yarl, urllib3, asynctest,
async-timeout, aiosignal, fsspec, aiohttp, xxhash, responses, datasets
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.24.3
    Uninstalling urllib3-1.24.3:
      Successfully uninstalled urllib3-1.24.3
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
datascience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is
incompatible.
Successfully installed aiohttp-3.8.1 aiosignal-1.2.0 async-timeout-4.0.2
asynctest-0.13.0 datasets-2.0.0 frozenlist-1.3.0 fsspec-2022.2.0 multidict-6.0.2
responses-0.18.0 urllib3-1.25.11 xxhash-3.0.0 yarl-1.7.2
```

We begin by querying Hugging Face to obtain the total count and names of the data sets. This

code obtains the total count and the names of the first five datasets.

[ ]:
```
from datasets import list_datasets

all_datasets = list_datasets()

print(f"Hugging Face hub currently contains {len(all_datasets)}")
print(f"datasets. The first 5 are:")
print("\n".join(all_datasets[:10]))
```

```
Hugging Face hub currently contains 3832
datasets. The first 5 are:
acronym_identification
ade_corpus_v2
adversarial_qa
aeslc
afrikaans_ner_corpus
ag_news
ai2_arc
air_dialogue
ajgt_twitter_ar
allegro_reviews
```

We begin by loading the emotion data set from the Hugging Face hub. Emotion is a dataset of English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise. [Cite:saravia2018carer] The following code loads the emotion data set from the Hugging Face hub.

[ ]:
```
from datasets import load_dataset

emotions = load_dataset("emotion")
```

```
Downloading builder script:   0%|          | 0.00/1.66k [00:00<?, ?B/s]

Downloading metadata:   0%|          | 0.00/1.61k [00:00<?, ?B/s]

Using custom data configuration default

Downloading and preparing dataset emotion/default (download: 1.97 MiB,
generated: 2.07 MiB, post-processed: Unknown size, total: 4.05 MiB) to /root/.ca
che/huggingface/datasets/emotion/default/0.0.0/348f63ca8e27b3713b6c04d723efe6d82
4a56fb3d1449794716c0f0296072705…

Downloading data:   0%|          | 0.00/1.66M [00:00<?, ?B/s]

Downloading data:   0%|          | 0.00/204k [00:00<?, ?B/s]

Downloading data:   0%|          | 0.00/207k [00:00<?, ?B/s]

Generating train split:   0%|          | 0/16000 [00:00<?, ? examples/s]

Generating validation split:   0%|          | 0/2000 [00:00<?, ? examples/s]

Generating test split:   0%|          | 0/2000 [00:00<?, ? examples/s]
```

```
Dataset emotion downloaded and prepared to /root/.cache/huggingface/datasets/emo
tion/default/0.0.0/348f63ca8e27b3713b6c04d723efe6d824a56fb3d1449794716c0f0296072
705. Subsequent calls will reuse this data.

  0%|          | 0/3 [00:00<?, ?it/s]
```

A quick scan of the downloaded data set reveals its structure. In this case, Hugging Face already separated the data into training, validation, and test data sets. The training set consists of 16,000 observations, while the test and validation sets contain 2,000 observations. The dataset is a Python dictionary that includes a Dataset object for each of these three divisions. The datasets only contain two columns, the text and the emotion label for each text sample.

[ ]:
```
emotions
```

[ ]:
```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 16000
    })
    validation: Dataset({
        features: ['text', 'label'],
        num_rows: 2000
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 2000
    })
})
```

You can see a single observation from the training data set here. This observation includes both the text sample and the assigned emotion label. The label is a numeric index representing the assigned emotion.

[ ]:
```
emotions['train'][2]
```

[ ]:
```
{'label': 3, 'text': 'im grabbing a minute to post i feel greedy wrong'}
```

We can display the labels in order of their index labels.

[ ]:
```
emotions['train'].features
```

[ ]:
```
{'label': ClassLabel(num_classes=6, names=['sadness', 'joy', 'love', 'anger',
'fear', 'surprise'], id=None),
 'text': Value(dtype='string', id=None)}
```

Hugging face can provide these data sets in a variety of formats. The following code receives the emotion data set as a Pandas data frame.

[ ]:
```
import pandas as pd
```

```
emotions.set_format(type='pandas')
df = emotions["train"][:]
df[:5]
```

[ ]:

	text	label
0	i didnt feel humiliated	0
1	i can go from feeling so hopeless to so damned…	0
2	im grabbing a minute to post i feel greedy wrong	3
3	i am ever feeling nostalgic about the fireplac…	2
4	i am feeling grouchy	3

We can use the Pandas "apply" function to add the textual label for each observation.

[ ]:
```
def label_it(row):
  return emotions["train"].features["label"].int2str(row)


df['label_name'] = df["label"].apply(label_it)
df[:5]
```

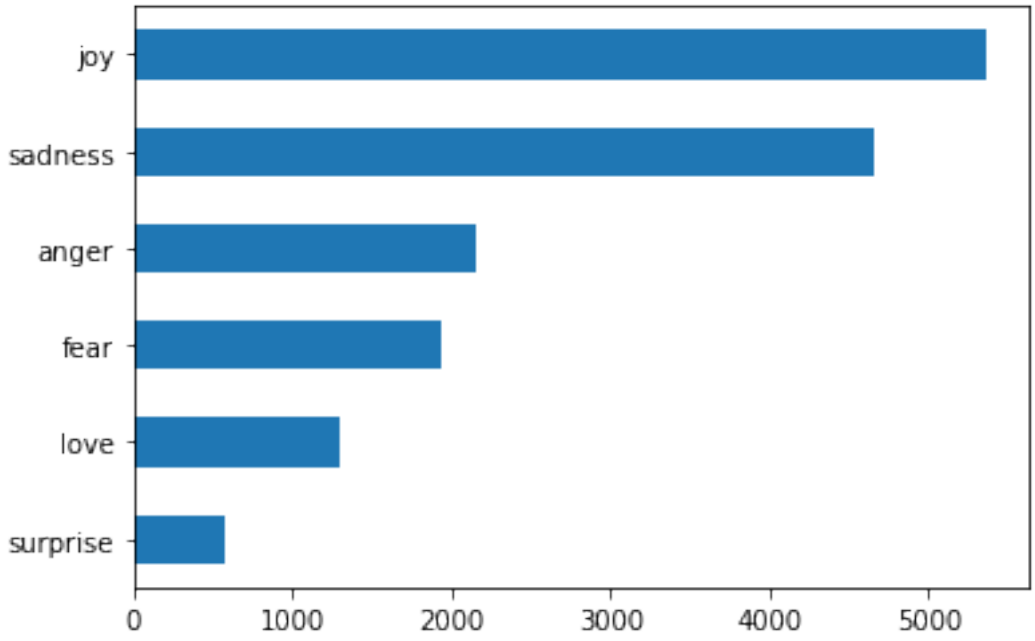
[ ]:

	text	label	label_name
0	i didnt feel humiliated	0	sadness
1	i can go from feeling so hopeless to so damned…	0	sadness
2	im grabbing a minute to post i feel greedy wrong	3	anger
3	i am ever feeling nostalgic about the fireplac…	2	love
4	i am feeling grouchy	3	anger

With the data in Pandas format and textually labeled, we can display a bar chart of the frequency of each of the emotions.

[ ]:
```
import matplotlib.pyplot as plt

df["label_name"].value_counts(ascending=True).plot.barh()
plt.show()
```



Finally, we utilize Hugging Face tokenizers and data sets together. The following code tokenizes the entire emotion data set. You can see below that the code has transformed the training set into subword tokens that are now ready to be used in conjunction with a transformer for either inference or training.

[ ]:
```
from transformers import AutoTokenizer


def tokenize(rows):
    return tokenizer(rows['text'], padding=True, truncation=True)


model_ckpt = "distilbert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_ckpt)

emotions.set_format(type=None)

encoded = tokenize(emotions["train"][:2])

print("**Input IDs**")
for a in encoded.input_ids:
    print(a)
```

```
Downloading:   0%|          | 0.00/28.0 [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/483 [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/226k [00:00<?, ?B/s]
```

```
Downloading:   0%|          | 0.00/455k [00:00<?, ?B/s]
```

```
**Input IDs**
[101, 1045, 2134, 2102, 2514, 26608, 102, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0]
[101, 1045, 2064, 2175, 2013, 3110, 2061, 20625, 2000, 2061, 9636, 17772, 2074,
2013, 2108, 2105, 2619, 2040, 14977, 1998, 2003, 8300, 102]
```

t81_558_class_11_04_hf_train

June 23, 2025

1 T81-558: Applications of Deep Neural Networks

Module 11: Natural Language Processing with Hugging Face * Instructor: Jeff Heaton, McKelvey School of Engineering, Washington University in St. Louis * For more information visit the class website.

2 Module 11 Material

- Part 11.1: Introduction to Hugging Face [Video] [Notebook]
- Part 11.2: Hugging Face Tokenizers [Video] [Notebook]
- Part 11.3: Hugging Face Datasets [Video] [Notebook]
- **Part 11.4: Training Hugging Face Models** [Video] [Notebook]
- Part 11.5: What are Embedding Layers in Keras [Video] [Notebook]

3 Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

[ ]:
```
try:
    %tensorflow_version 2.x
    COLAB = True
    print("Note: using Google CoLab")
except:
    print("Note: not using Google CoLab")
    COLAB = False
```

```
Note: using Google CoLab
```

4 Part 11.4: Training Hugging Face Models

Up to this point, we've used data and models from the Hugging Face hub unmodified. In this section, we will transfer and train a Hugging Face model. We will use Hugging Face data sets, tokenizers, and pretrained models to achieve this training.

We begin by installing Hugging Face if needed. It is also essential to install Hugging Face datasets.

[ ]:
```
# HIDE OUTPUT
!pip install transformers
```

```
!pip install transformers[sentencepiece]
!pip install datasets
```

```
Collecting transformers
  Downloading transformers-4.17.0-py3-none-any.whl (3.8 MB)
     |                | 3.8 MB 15.1 MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from transformers) (2.23.0)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.7/dist-packages (from transformers) (2019.12.20)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from transformers) (3.6.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from transformers) (1.21.5)
Collecting tokenizers!=0.11.3,>=0.11.1
  Downloading
tokenizers-0.11.6-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (6.5
MB)
     |                | 6.5 MB 56.8 MB/s
Collecting pyyaml>=5.1
  Downloading PyYAML-6.0-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (596 kB)
     |                | 596 kB 72.2 MB/s
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.7/dist-packages (from transformers) (21.3)
Collecting sacremoses
  Downloading sacremoses-0.0.49-py3-none-any.whl (895 kB)
     |                | 895 kB 65.0 MB/s
Requirement already satisfied: tqdm>=4.27 in
/usr/local/lib/python3.7/dist-packages (from transformers) (4.63.0)
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from transformers) (4.11.3)
Collecting huggingface-hub<1.0,>=0.1.0
  Downloading huggingface_hub-0.4.0-py3-none-any.whl (67 kB)
     |                | 67 kB 6.9 MB/s
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0,>=0.1.0->transformers) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging>=20.0->transformers)
(3.0.7)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->transformers) (3.7.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (3.0.4)
```

```
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->transformers) (2.10)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (7.1.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.1.0)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.15.0)
Installing collected packages: pyyaml, tokenizers, sacremoses, huggingface-hub,
transformers
  Attempting uninstall: pyyaml
    Found existing installation: PyYAML 3.13
    Uninstalling PyYAML-3.13:
      Successfully uninstalled PyYAML-3.13
Successfully installed huggingface-hub-0.4.0 pyyaml-6.0 sacremoses-0.0.49
tokenizers-0.11.6 transformers-4.17.0
Requirement already satisfied: transformers[sentencepiece] in
/usr/local/lib/python3.7/dist-packages (4.17.0)
Requirement already satisfied: sacremoses in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (0.0.49)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (3.6.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (4.63.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (21.3)
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(4.11.3)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (2.23.0)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(2019.12.20)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (6.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (1.21.5)
Requirement already satisfied: huggingface-hub<1.0,>=0.1.0 in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(0.4.0)
Requirement already satisfied: tokenizers!=0.11.3,>=0.11.1 in
/usr/local/lib/python3.7/dist-packages (from transformers[sentencepiece])
(0.11.6)
Collecting sentencepiece!=0.1.92,>=0.1.91
  Downloading
```

```
sentencepiece-0.1.96-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(1.2 MB)
     |                | 1.2 MB 14.4 MB/s
Requirement already satisfied: protobuf in /usr/local/lib/python3.7/dist-
packages (from transformers[sentencepiece]) (3.17.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0,>=0.1.0->transformers[sentencepiece]) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from
packaging>=20.0->transformers[sentencepiece]) (3.0.7)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->transformers[sentencepiece]) (3.7.0)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.7/dist-
packages (from protobuf->transformers[sentencepiece]) (1.15.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->transformers[sentencepiece]) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from
requests->transformers[sentencepiece]) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from
requests->transformers[sentencepiece]) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from
requests->transformers[sentencepiece]) (2021.10.8)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers[sentencepiece]) (7.1.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers[sentencepiece]) (1.1.0)
Installing collected packages: sentencepiece
Successfully installed sentencepiece-0.1.96
Collecting datasets
  Downloading datasets-2.0.0-py3-none-any.whl (325 kB)
     |                | 325 kB 14.5 MB/s
Collecting xxhash
  Downloading
xxhash-3.0.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (212 kB)
     |                | 212 kB 70.9 MB/s
Requirement already satisfied: multiprocess in
/usr/local/lib/python3.7/dist-packages (from datasets) (0.70.12.2)
Requirement already satisfied: huggingface-hub<1.0.0,>=0.1.0 in
/usr/local/lib/python3.7/dist-packages (from datasets) (0.4.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from datasets) (1.21.5)
Requirement already satisfied: requests>=2.19.0 in
/usr/local/lib/python3.7/dist-packages (from datasets) (2.23.0)
Requirement already satisfied: pyarrow>=5.0.0 in /usr/local/lib/python3.7/dist-
```

```
packages (from datasets) (6.0.1)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.7/dist-
packages (from datasets) (4.63.0)
Collecting responses<0.19
  Downloading responses-0.18.0-py3-none-any.whl (38 kB)
Requirement already satisfied: dill in /usr/local/lib/python3.7/dist-packages
(from datasets) (0.3.4)
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/dist-packages (from datasets) (4.11.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-
packages (from datasets) (21.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages
(from datasets) (1.3.5)
Collecting aiohttp
  Downloading aiohttp-3.8.1-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (1.1 MB)
     |                | 1.1 MB 62.9 MB/s
Collecting fsspec[http]>=2021.05.0
  Downloading fsspec-2022.2.0-py3-none-any.whl (134 kB)
     |                | 134 kB 74.3 MB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from huggingface-hub<1.0.0,>=0.1.0->datasets) (3.6.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages
(from huggingface-hub<1.0.0,>=0.1.0->datasets) (6.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.7/dist-packages (from huggingface-
hub<1.0.0,>=0.1.0->datasets) (3.10.0.2)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging->datasets) (3.0.7)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.19.0->datasets)
(1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.19.0->datasets) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests>=2.19.0->datasets) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.7/dist-packages (from requests>=2.19.0->datasets)
(2021.10.8)
Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1
  Downloading urllib3-1.25.11-py2.py3-none-any.whl (127 kB)
     |                | 127 kB 76.1 MB/s
Collecting asynctest==0.13.0
  Downloading asynctest-0.13.0-py3-none-any.whl (26 kB)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.7/dist-
packages (from aiohttp->datasets) (21.4.0)
Collecting multidict<7.0,>=4.5
```

```
  Downloading
multidict-6.0.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (94
kB)
     |                  | 94 kB 4.6 MB/s
Collecting yarl<2.0,>=1.0
  Downloading yarl-1.7.2-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_6
4.whl (271 kB)
     |                  | 271 kB 65.7 MB/s
Collecting frozenlist>=1.1.1
  Downloading frozenlist-1.3.0-cp37-cp37m-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (144 kB)
     |                  | 144 kB 78.0 MB/s
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.2.0-py3-none-any.whl (8.2 kB)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in
/usr/local/lib/python3.7/dist-packages (from aiohttp->datasets) (2.0.12)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata->datasets) (3.7.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
packages (from pandas->datasets) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.7/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-
packages (from python-dateutil>=2.7.3->pandas->datasets) (1.15.0)
Installing collected packages: multidict, frozenlist, yarl, urllib3, asynctest,
async-timeout, aiosignal, fsspec, aiohttp, xxhash, responses, datasets
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.24.3
    Uninstalling urllib3-1.24.3:
      Successfully uninstalled urllib3-1.24.3
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
datascience 0.10.6 requires folium==0.2.1, but you have folium 0.8.3 which is
incompatible.
Successfully installed aiohttp-3.8.1 aiosignal-1.2.0 async-timeout-4.0.2
asynctest-0.13.0 datasets-2.0.0 frozenlist-1.3.0 fsspec-2022.2.0 multidict-6.0.2
responses-0.18.0 urllib3-1.25.11 xxhash-3.0.0 yarl-1.7.2
```

We begin by loading the emotion data set from the Hugging Face hub. Emotion is a dataset of English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise. The following code loads the emotion data set from the Hugging Face hub.

```
[ ]: # HIDE OUTPUT
from datasets import load_dataset

emotions = load_dataset("emotion")
```

```
Downloading builder script:   0%|          | 0.00/1.66k [00:00<?, ?B/s]

Downloading metadata:   0%|          | 0.00/1.61k [00:00<?, ?B/s]

Using custom data configuration default

Downloading and preparing dataset emotion/default (download: 1.97 MiB,
generated: 2.07 MiB, post-processed: Unknown size, total: 4.05 MiB) to /root/.ca
che/huggingface/datasets/emotion/default/0.0.0/348f63ca8e27b3713b6c04d723efe6d82
4a56fb3d1449794716c0f0296072705…

Downloading data:   0%|          | 0.00/1.66M [00:00<?, ?B/s]

Downloading data:   0%|          | 0.00/204k [00:00<?, ?B/s]

Downloading data:   0%|          | 0.00/207k [00:00<?, ?B/s]

Generating train split:   0%|          | 0/16000 [00:00<?, ? examples/s]

Generating validation split:   0%|          | 0/2000 [00:00<?, ? examples/s]

Generating test split:   0%|          | 0/2000 [00:00<?, ? examples/s]

Dataset emotion downloaded and prepared to /root/.cache/huggingface/datasets/emo
tion/default/0.0.0/348f63ca8e27b3713b6c04d723efe6d824a56fb3d1449794716c0f0296072
705. Subsequent calls will reuse this data.

  0%|          | 0/3 [00:00<?, ?it/s]
```

You can see a single observation from the training data set here. This observation includes both the text sample and the assigned emotion label. The label is a numeric index representing the assigned emotion.

```
[ ]: emotions['train'][2]
```

```
[ ]: {'label': 3, 'text': 'im grabbing a minute to post i feel greedy wrong'}
```

We can display the labels in order of their index labels.

```
[ ]: emotions['train'].features
```

```
[ ]: {'label': ClassLabel(num_classes=6, names=['sadness', 'joy', 'love', 'anger',
     'fear', 'surprise'], id=None),
      'text': Value(dtype='string', id=None)}
```

Next, we utilize Hugging Face tokenizers and data sets together. The following code tokenizes the entire emotion data set. You can see below that the code has transformed the training set into subword tokens that are now ready to be used in conjunction with a transformer for either inference or training.

[ ]:
```
# HIDE OUTPUT
from transformers import AutoTokenizer


def tokenize(rows):
    return tokenizer(rows['text'], padding="max_length", truncation=True)


model_ckpt = "distilbert-base-uncased"
tokenizer = AutoTokenizer.from_pretrained(model_ckpt)

emotions.set_format(type=None)

tokenized_datasets = emotions.map(tokenize, batched=True)
```

```
Downloading:   0%|          | 0.00/28.0 [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/483 [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/226k [00:00<?, ?B/s]

Downloading:   0%|          | 0.00/455k [00:00<?, ?B/s]

  0%|          | 0/16 [00:00<?, ?ba/s]

  0%|          | 0/2 [00:00<?, ?ba/s]

  0%|          | 0/2 [00:00<?, ?ba/s]
```

We will utilize the Hugging Face **DefaultDataCollator** to transform the emotion data set into TensorFlow type data that we can use to finetune a neural network.

[ ]:
```
from transformers import DefaultDataCollator

data_collator = DefaultDataCollator(return_tensors="tf")
```

Now we generate a shuffled training and evaluation data set.

[ ]:
```
small_train_dataset = tokenized_datasets["train"].shuffle(seed=42)
small_eval_dataset = tokenized_datasets["test"].shuffle(seed=42)
```

We can now generate the TensorFlow data sets. We specify which columns should map to the input features and labels. We do not need to shuffle because we previously shuffled the data.

[ ]:
```
tf_train_dataset = small_train_dataset.to_tf_dataset(
    columns=["attention_mask", "input_ids", "token_type_ids"],
    label_cols=["labels"],
    shuffle=True,
    collate_fn=data_collator,
    batch_size=8,
)
```

```
tf_validation_dataset = small_eval_dataset.to_tf_dataset(
    columns=["attention_mask", "input_ids", "token_type_ids"],
    label_cols=["labels"],
    shuffle=False,
    collate_fn=data_collator,
    batch_size=8,
)
```

We will now load the distilbert model for classification. We will adjust the pretrained weights to predict the emotions of text lines.

[ ]:
```
# HIDE OUTPUT
import tensorflow as tf
from transformers import TFAutoModelForSequenceClassification

model = TFAutoModelForSequenceClassification.from_pretrained(\
    "distilbert-base-uncased", num_labels=6)
```

```
Downloading:   0%|          | 0.00/347M [00:00<?, ?B/s]
```

```
Some layers from the model checkpoint at distilbert-base-uncased were not used
when initializing TFDistilBertForSequenceClassification: ['vocab_layer_norm',
'vocab_transform', 'vocab_projector', 'activation_13']
- This IS expected if you are initializing TFDistilBertForSequenceClassification
from the checkpoint of a model trained on another task or with another
architecture (e.g. initializing a BertForSequenceClassification model from a
BertForPreTraining model).
- This IS NOT expected if you are initializing
TFDistilBertForSequenceClassification from the checkpoint of a model that you
expect to be exactly identical (initializing a BertForSequenceClassification
model from a BertForSequenceClassification model).
Some layers of TFDistilBertForSequenceClassification were not initialized from
the model checkpoint at distilbert-base-uncased and are newly initialized:
['pre_classifier', 'classifier', 'dropout_19']
You should probably TRAIN this model on a down-stream task to be able to use it
for predictions and inference.
```

We now train the neural network. Because the network is already pretrained, we use a small learning rate.

[ ]:
```
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=5e-5),
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=tf.metrics.SparseCategoricalAccuracy(),
)

model.fit(tf_train_dataset, validation_data=tf_validation_dataset,
          epochs=5)
```

```
Epoch 1/5
```

```
2000/2000 [==============================] - 360s 174ms/step - loss: 0.3720 -
sparse_categorical_accuracy: 0.8669 - val_loss: 0.1728 -
val_sparse_categorical_accuracy: 0.9180
Epoch 2/5
2000/2000 [==============================] - 347s 174ms/step - loss: 0.1488 -
sparse_categorical_accuracy: 0.9338 - val_loss: 0.1496 -
val_sparse_categorical_accuracy: 0.9295
Epoch 3/5
2000/2000 [==============================] - 347s 173ms/step - loss: 0.1253 -
sparse_categorical_accuracy: 0.9420 - val_loss: 0.1617 -
val_sparse_categorical_accuracy: 0.9245
Epoch 4/5
2000/2000 [==============================] - 346s 173ms/step - loss: 0.1092 -
sparse_categorical_accuracy: 0.9486 - val_loss: 0.1654 -
val_sparse_categorical_accuracy: 0.9295
Epoch 5/5
2000/2000 [==============================] - 347s 173ms/step - loss: 0.0960 -
sparse_categorical_accuracy: 0.9585 - val_loss: 0.1830 -
val_sparse_categorical_accuracy: 0.9220
```

[ ]: <keras.callbacks.History at 0x7f42e84a49d0>

Open in Colab

T81-558: Applications of Deep Neural Networks

Module 11: Natural Language Processing and Speech Recognition

- Instructor: Jeff Heaton, McKelvey School of Engineering, Washington University in St. Louis
- For more information visit the class website.

Module 11 Material

- Part 11.1: Introduction to Hugging Face [Video] [Notebook]
- Part 11.2: Hugging Face Tokenizers [Video] [Notebook]
- Part 11.3: Hugging Face Datasets [Video] [Notebook]
- Part 11.4: Training Hugging Face Models [Video] [Notebook]
- **Part 11.5: What are Embedding Layers in Keras** [Video] [Notebook]

Google CoLab Instructions

The following code ensures that Google CoLab is running the correct version of TensorFlow.

In [ ]:
```
try:
    %tensorflow_version 2.x
    COLAB = True
    print("Note: using Google CoLab")
except:
    print("Note: not using Google CoLab")
    COLAB = False
```

```
Note: using Google CoLab
```

Part 11.5: What are Embedding Layers in Keras

Embedding Layers are a handy feature of Keras that allows the program to automatically insert additional information into the data flow of your neural network. In the previous section, you saw that Word2Vec could expand words to a 300

dimension vector. An embedding layer would automatically allow you to insert these 300-dimension vectors in the place of word indexes.

Programmers often use embedding layers with Natural Language Processing (NLP); however, you can use these layers when you wish to insert a lengthier vector in an index value place. In some ways, you can think of an embedding layer as dimension expansion. However, the hope is that these additional dimensions provide more information to the model and provide a better score.

Simple Embedding Layer Example

- **input_dim** = How large is the vocabulary? How many categories are you encoding? This parameter is the number of items in your "lookup table."
- **output_dim** = How many numbers in the vector you wish to return.
- **input_length** = How many items are in the input feature vector that you need to transform?

Now we create a neural network with a vocabulary size of 10, which will reduce those values between 0-9 to 4 number vectors. This neural network does nothing more than passing the embedding on to the output. But it does let us see what the embedding is doing. Each feature vector coming in will have two such features.

In [ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding
import numpy as np

model = Sequential()
embedding_layer = Embedding(input_dim=10, output_dim=4, input_length=2)
model.add(embedding_layer)
model.compile('adam', 'mse')
```

Let's take a look at the structure of this neural network to see what is happening inside it.

In [ ]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 2, 4)              40

=================================================================
Total params: 40
Trainable params: 40
Non-trainable params: 0
_________________________________________________________________
```

For this neural network, which is just an embedding layer, the input is a vector of size 2. These two inputs are integer numbers from 0 to 9 (corresponding to the requested input_dim quantity of 10 values). Looking at the summary above, we see that the embedding layer has 40 parameters. This value comes from the embedded lookup table that contains four amounts (output_dim) for each of the 10 (input_dim) possible integer values for the two inputs. The output is 2 (input_length) length 4 (output_dim) vectors, resulting in a total output size of 8, which corresponds to the Output Shape given in the summary above.

Now, let us query the neural network with two rows. The input is two integer values, as was specified when we created the neural network.

In [ ]:
```
input_data = np.array([
    [1, 2]
])

pred = model.predict(input_data)

print(input_data.shape)
print(pred)
```

```
(1, 2)
[[[-0.04494917  0.01937468 -0.00152863  0.04808659]
  [-0.04002655  0.03441895  0.04462588 -0.01472597]]]
```

Here we see two length-4 vectors that Keras looked up for each input integer. Recall that Python arrays are zero-based. Keras replaced the value of 1 with the second row of the 10 x 4 lookup matrix. Similarly, Keras returned the value of 2 by the third row of the lookup matrix. The following code displays the lookup matrix in its entirety. The embedding layer performs no mathematical operations other than inserting the correct row from the lookup table.

In [ ]:
```
embedding_layer.get_weights()
```

Out[ ]:
```
[array([[-0.03164196,  0.02898774, -0.0273805 ,  0.01066511],
        [-0.04494917,  0.01937468, -0.00152863,  0.04808659],
        [-0.04002655,  0.03441895,  0.04462588, -0.01472597],
        [ 0.02480464, -0.02585896,  0.0099823 ,  0.02589831],
        [-0.02502655,  0.02517617, -0.03199299,  0.00127842],
        [-0.00205797,  0.02709344, -0.04335414, -0.01793201],
        [ 0.03926537,  0.0293855 ,  0.0445295 , -0.02160555],
        [-0.0075082 , -0.03241253,  0.04906586, -0.02384975],
        [ 0.00264529, -0.01921672, -0.0031809 ,  0.00151991],
        [-0.02407705, -0.04659952, -0.02667597, -0.04108504]],
       dtype=float32)]
```

The values above are random parameters that Keras generated as starting points. Generally, we will transfer an embedding or train these random values into something useful. The following section demonstrates how to embed a hand-coded embedding.

Transferring An Embedding

Now, we see how to hard-code an embedding lookup that performs a simple one-hot encoding. One-hot encoding would transform the input integer values of 0, 1, and 2 to the vectors $[1, 0, 0]$, $[0, 1, 0]$, and $[0, 0, 1]$ respectively. The following code replaced the random lookup values in the embedding layer with this one-hot coding-inspired lookup table.

In [ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding
import numpy as np

embedding_lookup = np.array([
    [1, 0, 0],
    [0, 1, 0],
    [0, 0, 1]
])

model = Sequential()
embedding_layer = Embedding(input_dim=3, output_dim=3, input_length=2)
model.add(embedding_layer)
model.compile('adam', 'mse')

embedding_layer.set_weights([embedding_lookup])
```

We have the following parameters for the Embedding layer:

- input_dim=3 - There are three different integer categorical values allowed.
- output_dim=3 - Three columns represent a categorical value with three possible values per one-hot encoding.
- input_length=2 - The input vector has two of these categorical values.

We query the neural network with two categorical values to see the lookup performed.

In [ ]:
```
input_data = np.array([
    [0, 1]
])

pred = model.predict(input_data)

print(input_data.shape)
print(pred)
```

```
(1, 2)
[[[1. 0. 0.]
  [0. 1. 0.]]]
```

The given output shows that we provided the program with two rows from the one-hot encoding table. This encoding is a correct one-hot encoding for the values 0 and 1, where there are up to 3 unique values possible.

The following section demonstrates how to train this embedding lookup table.

Training an Embedding

First, we make use of the following imports.

In [ ]:
```
from numpy import array
from tensorflow.keras.preprocessing.text import one_hot
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Flatten, Embedding, Dense
```

We create a neural network that classifies restaurant reviews according to positive or negative. This neural network can accept strings as input, such as given here. This code also includes positive or negative labels for each review.

In [ ]:
```
# Define 10 resturant reviews.
reviews = [
    'Never coming back!',
    'Horrible service',
    'Rude waitress',
    'Cold food.',
    'Horrible food!',
    'Awesome',
    'Awesome service!',
    'Rocks!',
    'poor work',
    'Couldn\'t have done better']

# Define labels (1=negative, 0=positive)
labels = array([1, 1, 1, 1, 1, 0, 0, 0, 0, 0])
```

Notice that the second to the last label is incorrect. Errors such as this are not too out of the ordinary, as most training data could have some noise.

We define a vocabulary size of 50 words. Though we do not have 50 words, it is okay to use a value larger than needed. If there are more than 50 words, the least frequently used words in the training set are automatically dropped by the embedding layer during training. For input, we one-hot encode the strings. We use the TensorFlow one-hot encoding method here rather than Scikit-Learn. Scikit-learn would expand these strings to the 0's and 1's as we would typically see for dummy variables. TensorFlow translates all words to index values and replaces each word with that index.

In [ ]:
```
VOCAB_SIZE = 50
encoded_reviews = [one_hot(d, VOCAB_SIZE) for d in reviews]
print(f"Encoded reviews: {encoded_reviews}")
```

```
Encoded reviews: [[40, 43, 7], [27, 31], [49, 46], [2, 28], [27, 28], [20],
[20, 31], [39], [18, 39], [11, 3, 18, 11]]
```

The program one-hot encodes these reviews to word indexes; however, their lengths are different. We pad these reviews to 4 words and truncate any words beyond the fourth word.

In [ ]:
```
MAX_LENGTH = 4

padded_reviews = pad_sequences(encoded_reviews, maxlen=MAX_LENGTH,
                               padding='post')
print(padded_reviews)
```

```
[[40 43  7  0]
 [27 31  0  0]
 [49 46  0  0]
 [ 2 28  0  0]
 [27 28  0  0]
 [20  0  0  0]
 [20 31  0  0]
 [39  0  0  0]
 [18 39  0  0]
 [11  3 18 11]]
```

As specified by the **padding=post** setting, each review is padded by appending zeros at the end, as specified by the **padding=post** setting.

Next, we create a neural network to learn to classify these reviews.

In [ ]:
```
model = Sequential()
embedding_layer = Embedding(VOCAB_SIZE, 8, input_length=MAX_LENGTH)
model.add(embedding_layer)
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy',
              metrics=['acc'])

print(model.summary())
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 4, 8)              400

 flatten (Flatten)           (None, 32)                0

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 433
Trainable params: 433
Non-trainable params: 0
_________________________________________________________________
None
```

This network accepts four integer inputs that specify the indexes of a padded movie review. The first embedding layer converts these four indexes into four length vectors 8. These vectors come from the lookup table that contains 50 (VOCAB_SIZE) rows of vectors of length 8. This encoding is evident by the 400 (8 times 50) parameters in the embedding layer. The output size from the embedding layer is 32 (4 words expressed as 8-number embedded vectors). A single output neuron is connected to the embedding layer by 33 weights (32 from the embedding layer and a single bias neuron). Because this is a single-class classification network, we use the sigmoid activation function and binary_crossentropy.

The program now trains the neural network. The embedding lookup and dense 33 weights are updated to produce a better score.

In [ ]:
```
# fit the model
model.fit(padded_reviews, labels, epochs=100, verbose=0)
```

Out[ ]:
```
<keras.callbacks.History at 0x7fd87794c4d0>
```

We can see the learned embeddings. Think of each word's vector as a location in the 8 dimension space where words associated with positive reviews are close to other words. Similarly, training places negative reviews close to each other. In addition to the training setting these embeddings, the 33 weights between the embedding layer and output neuron similarly learn to transform these embeddings into an actual prediction. You can see these embeddings here.

In [ ]:
```
print(embedding_layer.get_weights()[0].shape)
print(embedding_layer.get_weights())
```

```
(50, 8)
[array([[-0.11389559, -0.04778124,  0.10034387,  0.12887037,  0.05670259,
         -0.09982903, -0.15423775, -0.06774805],
        [-0.04839246,  0.00527745,  0.0084306 , -0.03498586,  0.010772  ,
          0.04015711,  0.03564452, -0.00849336],
        [-0.11003157, -0.05829103,  0.12370535, -0.07124459, -0.0667479 ,
         -0.14339209, -0.13791779, -0.13947721],
        [-0.15395765, -0.08560142, -0.15915371, -0.0882007 ,  0.15756004,
         -0.10337664, -0.12412377, -0.10282961],
        [ 0.04919637, -0.00870635, -0.02393281,  0.04445953,  0.0124351 ,
          0.02354855, -0.02476437,  0.04543931],
        [-0.00503131,  0.01302261, -0.02866241,  0.04487506, -0.04427315,
          0.00651342, -0.02796236,  0.03458978],
        [-0.03738759, -0.00135366,  0.04961893, -0.04076886, -0.0007545 ,
          0.03454826,  0.03419926, -0.00689811],
        [ 0.14487585,  0.14052217, -0.08246625, -0.08622362,  0.10270283,
         -0.06439426, -0.16649802, -0.11733696],
        [-0.01337775,  0.00189237, -0.04226214, -0.02981731, -0.04849073,
          0.0464913 , -0.04499427, -0.04841725],
        [-0.01929135, -0.02657523, -0.0335291 ,  0.04808146,  0.02409947,
         -0.03780599,  0.03453754,  0.00598647],
        [ 0.03076488, -0.03929596,  0.00840779, -0.03980947,  0.04209021,
         -0.00642526,  0.03741593,  0.04605447],
        [ 0.11537231, -0.10763969, -0.06139125,  0.07191044,  0.05322507,
          0.15153708, -0.14278722,  0.11250742],
        [-0.04048342, -0.02535482, -0.01568266, -0.02351468,  0.00865855,
          0.04086712, -0.03859865,  0.0365578 ],
        [-0.0009298 , -0.0311846 , -0.03491043, -0.00289371,  0.00757905,
         -0.03187181, -0.02323085, -0.01488547],
        [ 0.0320026 ,  0.03818611,  0.00219003, -0.03297286, -0.03609738,
         -0.00905116, -0.00735079, -0.0369678 ],
        [ 0.04876169,  0.04988963, -0.01918377,  0.02061111, -0.03650783,
          0.00809064,  0.00043495, -0.02308334],
        [-0.02140537,  0.02220272,  0.00469884,  0.0342283 ,  0.01847946,
          0.02940113, -0.04855499,  0.02044804],
        [-0.00828004, -0.0079689 ,  0.01667002,  0.0414703 , -0.01305557,
          0.04526286, -0.01467935,  0.01147614],
        [-0.14282468, -0.08361981, -0.11100344,  0.1147782 ,  0.13931683,
          0.05983332,  0.16483088,  0.09642172],
        [-0.04617438,  0.04929153,  0.0485074 , -0.02250378,  0.01294557,
         -0.0425485 , -0.01274359,  0.00403596],
        [ 0.08578632,  0.10722891, -0.10169367,  0.05640666,  0.13935997,
          0.07905768,  0.0912255 ,  0.14614286],
        [-0.02422597, -0.02895569,  0.02458526, -0.02941357,  0.03783615,
          0.0217586 ,  0.04737884,  0.03385517],
        [-0.01605659,  0.02846745, -0.04217149,  0.00933688, -0.015615  ,
         -0.0185383 ,  0.03455376,  0.0217413 ],
        [-0.02496419, -0.01964381, -0.01747011, -0.0086274 , -0.00279769,
         -0.00473202,  0.04959089, -0.02818167],
        [-0.01308316,  0.0437695 , -0.01201218, -0.00937818, -0.03936937,
          0.03369248,  0.01404865,  0.01300433],
        [-0.03047577, -0.04215126, -0.03603753, -0.01572833,  0.04595536,
         -0.01445602,  0.02598487, -0.03712183],
        [ 0.04174629,  0.030602  , -0.01565778,  0.01411921, -0.03829115,
          0.02699218, -0.03978034, -0.00037332],
        [-0.05509803, -0.12121415,  0.12930614, -0.14208739, -0.05467908,
```

```
 -0.10421305, -0.1347957 , -0.09714746],
[ 0.14368567,  0.14523256,  0.15996216,  0.07271292, -0.10887505,
  0.07155557,  0.10750765,  0.14647684],
[-0.04667553, -0.00594231, -0.04209081, -0.01220823, -0.02044651,
  0.02359882,  0.01033651, -0.01691378],
[ 0.02788267, -0.0466502 , -0.04354659, -0.04944308,  0.00530468,
  0.03017677,  0.01628789,  0.00456915],
[ 0.09592342,  0.05642203,  0.03576508,  0.06546731, -0.03308697,
  0.03154759,  0.00280966,  0.03369548],
[-0.00399817, -0.02812622, -0.00763954, -0.003208  ,  0.04371027,
 -0.03186812,  0.01646887, -0.04135863],
[ 0.00120915,  0.00111195,  0.01940939,  0.0100676 ,  0.02689103,
 -0.02420806,  0.04829462, -0.00500059],
[-0.00374997,  0.00533805,  0.01584294, -0.01231242, -0.02583057,
 -0.00426785, -0.01593303,  0.03316021],
[-0.00542512, -0.02522955,  0.01944559,  0.04694534,  0.01956921,
 -0.04743705,  0.01203604, -0.04249186],
[ 0.04021386, -0.00147871, -0.03729609, -0.04367838, -0.02620382,
 -0.03366937, -0.04764401,  0.01843042],
[-0.04885202, -0.04030935, -0.02691921, -0.04069231,  0.00133073,
  0.04187706,  0.01700257, -0.0269224 ],
[-0.04759267, -0.02806743, -0.02340071,  0.04413268,  0.04873205,
 -0.02571398,  0.02112493,  0.01220033],
[ 0.03645799,  0.04670727, -0.14964601,  0.06317957,  0.12738568,
 -0.05583218, -0.07265829, -0.11887868],
[-0.0461492 , -0.14710744,  0.14215472, -0.08502222, -0.11263344,
 -0.10313905, -0.09941045, -0.0613514 ],
[-0.01235803, -0.03596945,  0.04333005, -0.02633744,  0.0076986 ,
 -0.02331397, -0.02244077,  0.02170218],
[ 0.02890852, -0.02253481, -0.04383245, -0.00917351,  0.01134578,
  0.0413558 , -0.00813813,  0.03958623],
[ 0.13829918,  0.0676541 ,  0.16875601,  0.04536283, -0.12547925,
  0.13549416,  0.06408142,  0.1365626 ],
[ 0.02720174,  0.02317807, -0.01934367,  0.03661523, -0.00081351,
 -0.00664594, -0.01546872,  0.00292607],
[ 0.03418565, -0.02236365, -0.03703803,  0.01724467, -0.02788099,
 -0.01143361, -0.00885036, -0.00753104],
[ 0.11629202,  0.08401583,  0.12823549,  0.04578856, -0.10711329,
  0.12236115,  0.12761551,  0.12674938],
[-0.01328101,  0.01608239, -0.02894524,  0.03419088,  0.04457787,
  0.02493219,  0.04973162,  0.03453101],
[-0.00029699, -0.0425287 ,  0.02509956, -0.00861088,  0.04153964,
 -0.04445877, -0.00612149, -0.03430663],
[-0.08493928, -0.10910758,  0.0605178 , -0.10072854, -0.11677803,
 -0.05648913, -0.13342443, -0.08516318]], dtype=float32)]
```

We can now evaluate this neural network's accuracy, including the embeddings and the learned dense layer.

In [ ]:
```
loss, accuracy = model.evaluate(padded_reviews, labels, verbose=0)
print(f'Accuracy: {accuracy}')
```

```
Accuracy: 1.0
```

The accuracy is a perfect 1.0, indicating there is likely overfitting. It would be good to use early stopping to not overfit for a more complex data set.

In [ ]:
```
print(f'Log-loss: {loss}')
```

```
Log-loss: 0.48446863889694214
```

However, the loss is not perfect. Even though the predicted probabilities indicated a correct prediction in every case, the program did not achieve absolute confidence in each correct answer. The lack of confidence was likely due to the small amount of noise (previously discussed) in the data set. Some words that appeared in both positive and negative reviews contributed to this lack of absolute certainty.

In [ ]:
```

```