Aula_8_BERT_(Rafael_Ito)

April 29, 2020

Sentiment analysis using BERT

Author: Rafael Ito

e-mail: ito.rafael@gmail.com

0. Dataset and Description

Name: IMDb

Description: this notebook uses the IMDb dataset which contains movie reviews classified as either positive or negative review. The aim is to perform a supervised learning for sentiment classification using the BERT model.

1. Libraries and packages

1.1 Install packages

```
[1]: !pip install -q
         numpy
         torch
         sklearn
         skorch
         matplotlib \
         pytorch-lightning
         transformers
         || 122kB 2.8MB/s
         || 235kB 8.7MB/s
         || 573kB 46.0MB/s
         || 3.7MB 34.9MB/s
         || 890kB 45.5MB/s
         || 1.0MB 44.1MB/s
      Building wheel for sacremoses (setup.py) ... done
    ERROR: pytorch-lightning 0.7.5 has requirement future>=0.17.1, but you'll
    have future 0.16.0 which is incompatible.
    ERROR: pytorch-lightning 0.7.5 has requirement tqdm>=4.41.0, but you'll
    have tqdm 4.38.0 which is incompatible.
```

1.2 Import libraries

```
[2]: #-----
    # general
    import numpy as np
    import pandas as pd
    import os
    import random
    import itertools
    import collections
    from argparse import Namespace
    from typing import Dict
    from typing import List
    from multiprocessing import cpu_count
    import tensorboard
    %load_ext tensorboard
    #----
    # NLP
    import re
    import nltk
    from transformers import BertTokenizer
    from transformers import BertModel
    # PyTorch
    import torch
    from torch.nn import CrossEntropyLoss, MSELoss
    from torch.nn import Linear
    from torch.optim import Adam, SGD
    from torch.optim.lr_scheduler import StepLR
    from torch.utils.data import TensorDataset, Dataset, DataLoader
    import torch.nn.functional as F
    #-----
    # PyTorch Lightning
    import pytorch_lightning as pl
    from pytorch_lightning.callbacks import EarlyStopping, ModelCheckpoint
    from pytorch_lightning.loggers import TensorBoardLogger
    # scikit-learn
    from sklearn.metrics import \
        confusion_matrix, accuracy_score, precision_score, f1_score
    # data visualization
    import matplotlib.pyplot as plt
    import seaborn as sns
                         _____
    # additional config
    #-----
    # random seed generator
    np.random.seed(42)
    torch.manual_seed(42);
    print('Torch version:', torch.__version__)
    print('Pytorch Lightning version:', pl.__version__)
```

```
Torch version: 1.5.0+cu101
Pytorch Lightning version: 0.7.5

/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

```
[0]: # the next function is based on the reference notebook from Diedre.
```

```
[4]: # library and function to monitor GPU usage
    # (which should be always close to 100% during the training phase)

import nvidia_smi
nvidia_smi.nvmlInit()
handle = nvidia_smi.nvmlDeviceGetHandleByIndex(0)
print(f'Device name: {nvidia_smi.nvmlDeviceGetName(handle)}')

def gpu_usage():
    global handle
    return str(nvidia_smi.nvmlDeviceGetUtilizationRates(handle).gpu) + '%'
```

Device name: b'Tesla P100-PCIE-16GB'

1.3 Check device

Device: cuda
GPU model: Tesla P100-PCIE-16GB
GPU memory: 17.07 GB
#----CPU cores: 4

1.4 Constants definition

[0]:

2. Custom functions and classes

2.1 Functions

Function that calculates the number of parameters of a network

```
[O]:

description:

- given a model, this function returns its number of parameters (weight, bias)

#-----

positional args:

- model [torch.nn.Module]: instance of the network

optional args:

- verbose (default=False) [bool]: if True, print a report with the parameters of □

→ each layer

- all_parameters (default=False) [bool]:

if True, return number of all parameters, if False, return only trainable □

→ parameters

#-----

return:

- [int] total parameters of the network

''';
```

```
[0]: def nparam(model, verbose=False, all_parameters=False):
         if(verbose):
             i = 0
             total = 0
             for name, param in model.named_parameters():
                 if (param.requires_grad):
                     #print('layer ', i, ' name: ', name)
                     j = 1
                     for dim in param.data.shape:
                         j = j * dim
                     print('layer ', i, ': ', name, '; parameters: ', j, sep='')
                     i += 1
                     total += j
             print('total parameters = ', total)
             return
         else:
             if (all_parameters):
                 return sum(p.numel() for p in model.parameters())
                 return sum(p.numel() for p in model.parameters() if p.requires_grad)
```

Function to plot confusion matrix

```
[0]:

description:

this function plots the confusion matrix (normalized or not)
using Matplotlib and seaborn in a nice way using heatmap.

#------

positional args:

confusion_matrix [numpy.ndarray]: ex.: array([[88, 19],[22, 71]])

class_names [list of str]: ex.: ['negative', 'positive']

optional args:

title (default=None) [str]: title of the plot

normalize (default=False) [bool]: values raw or normalized

cmap (default=plt.cm.Blues)
```

```
[matplotlib.colors.LinearSegmentedColormap]: colormap to be used
  - fig_size (default=(10,7)) [tuple]: size of the figure
  - fontsize (default=14) [int]: size of the text
#--------
return:
  - fig [matplotlib.figure.Figure]: confusion matrix plotted in a nice way!
''';
```

```
[0]: #https://github.com/ito-rafael/machine-learning/blob/master/snippets/confusion_matrix.
     def print_confusion_matrix(confusion_matrix, class_names, title=None, normalize=False,_u
      →cmap=plt.cm.Blues, figsize = (10,7), fontsize=14):
         # normalized or raw CM
         if normalize:
             confusion_matrix = confusion_matrix.astype('float') / confusion_matrix.
      →sum(axis=1)[:, np.newaxis]
             fmt = '.2f'
         else:
             fmt = 'd'
         df_cm = pd.DataFrame(confusion_matrix, index=class_names, columns=class_names)
         fig = plt.figure(figsize=figsize)
         try:
             heatmap = sns.heatmap(df_cm, annot=True, fmt=fmt, cmap=cmap)
         except ValueError:
             raise ValueError("Confusion matrix values must be integers.")
         # fix matplotlib 3.1.1 bug
         \#heatmap.get\_ylim() --> (5.5, 0.5)
         \#heatmap.set\_ylim(6.0, 0)
         heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0,_u
      →ha='right', fontsize=fontsize)
         heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=45,__
      →ha='right', fontsize=fontsize)
         plt.title(title)
         plt.ylabel('True label')
         plt.xlabel('Predicted label')
         return fig
```

Function that preprocess a document, returning the mean of word embeddings

```
[0]:

description:

this function receives as parameter a corpus [list of lists] and do the following:

- convert to lower case,

- split in tokens,

- remove stop words

return:

the same corpus preprocessed

''';
```

```
[0]: def pre_processing(corpus, stopwords, embedding):
         corpus_pp = []
         for sentence in corpus:
             sentence = sentence.lower()
                                                         # convert to lower case
             sentence = re.sub("[^\w]", " ", sentence) # match word characters_
      \rightarrow [a-zA-Z0-9]
             sentence = sentence.split()
                                                          # split in tokens
             sentence_pp = []
             for token in sentence:
                 # remove stop words
                 if token not in stopwords:
                     sentence_pp.append(token)
             corpus_pp.append(sentence_pp)
         return corpus_pp
```

2.2 Classes

[0]:

3. Dataset Pre-processing

3.1 Download dataset

```
[12]: # download complete dataset (50k samples: 25k train, 25k test)
!wget -nc http://files.fast.ai/data/aclImdb.tgz
!tar -xzf aclImdb.tgz

--2020-04-29 18:02:09-- http://files.fast.ai/data/aclImdb.tgz
Resolving files.fast.ai (files.fast.ai)... 67.205.15.147
Connecting to files.fast.ai (files.fast.ai)|67.205.15.147|:80... connected.
HTTP request sent, awaiting response... 200 0K
Length: 145982645 (139M) [application/x-gtar-compressed]
Saving to: 'aclImdb.tgz'

aclImdb.tgz 100%[============] 139.22M 95.7MB/s in 1.5s
2020-04-29 18:02:11 (95.7 MB/s) - 'aclImdb.tgz' saved [145982645/145982645]
```

3.2 Dataset Class

```
BERT input:
[CLS] + tokens + [SEP] + padding
```

```
self.texts = texts
      self.labels = torch.LongTensor(labels)
      self.max_length = max_length
       # special tokens
      self.CLS_ID = [self.tokenizer.cls_token_id]
       self.SEP_ID = [self.tokenizer.sep_token_id]
      self.PAD_ID = [self.tokenizer.pad_token_id]
  def __len__(self):
      return len(self.labels)
  def __getitem__(self, idx):
      text = self.texts[idx]
       # tokenize text
      tokens = self.tokenizer.tokenize(text)
       # truncate
      tokens_trunc = tokens[:(self.max_length - 2)]
       # convert do ids
      word_ids = self.tokenizer.convert_tokens_to_ids(tokens_trunc)
       # create mask
      attention_mask = [1] * (len(word_ids) + 2) + [0] * (self.max_length -\Box
\rightarrow (len(word_ids) + 2))
       # create token type ID
      token_type_id = torch.zeros(self.max_length, dtype=torch.int64)
       # complete with pad
      token_ids = self.CLS_ID + word_ids + self.SEP_ID + \
           self.PAD_ID * (self.max_length - (len(word_ids) + 2))
      return torch.LongTensor(token_ids), torch.LongTensor(attention_mask),_
→token_type_id, self.labels[idx]
```

3.3 Dataloader testing

```
[14]: tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
```

INFO:filelock:Lock 140253302421432 acquired on /root/.cache/torch/transformers/2 6bc1ad6c0ac742e9b52263248f6d0f00068293b33709fae12320c0e35ccfbbb.542ce4285a40d23a 559526243235df47c5f75c197f04f37d1a0c124c32c9a084.lock

INFO:transformers.file_utils:https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-vocab.txt not found in cache or force_download set to True, downloading to /root/.cache/torch/transformers/tmp7fju0n3g

HBox(children=(IntProgress(value=0, description='Downloading', max=231508, style=ProgressStyle(descript

```
INFO:transformers.file_utils:storing
```

 $\label{local_https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-vocab.txt in cache at /root/.cache/torch/transformers/26bc1ad6c0ac742e9b52263248f6d0f00068 293b33709fae12320c0e35ccfbbb.542ce4285a40d23a559526243235df47c5f75c197f04f37d1a0 c124c32c9a084$

 $INFO: transformers. file_utils: creating metadata file for /root/.cache/torch/transformers/26bc1ad6c0ac742e9b52263248f6d0f00068293b33709fae12320c0e35ccfbbb.542ce4285a40d23a559526243235df47c5f75c197f04f37d1a0c124c32c9a084$

INFO:filelock:Lock 140253302421432 released on /root/.cache/torch/transformers/2 6bc1ad6c0ac742e9b52263248f6d0f00068293b33709fae12320c0e35ccfbbb.542ce4285a40d23a

```
559526243235df47c5f75c197f04f37d1a0c124c32c9a084.lock INFO:transformers.tokenization_utils:loading file
```

 $\label{local-states} $$ $$ $ \text{https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-vocab.txt from cache at /root/.cache/torch/transformers/26bc1ad6c0ac742e9b52263248f6d0f000 68293b33709fae12320c0e35ccfbbb.542ce4285a40d23a559526243235df47c5f75c197f04f37d1 a0c124c32c9a084}$

```
[15]: texts = ['we like pizza', 'he does not like apples']
     labels = [0, 1]
     dataset_debug = IMDbDataset(
        texts=texts,
        labels=labels,
        tokenizer=tokenizer,
        max_length=20)
     dataloader_debug = DataLoader(dataset_debug, batch_size=10, shuffle=True,
                              num_workers=0)
     token_ids, attention_mask, token_type_ids, labels = next(iter(dataloader_debug))
     print('token_ids:\n', token_ids)
     print('token_type_ids:\n', token_type_ids)
     print('attention_mask:\n', attention_mask)
     print('labels:\n', labels)
     print('#----')
     print('token_ids.shape:', token_ids.shape)
     print('token_type_ids.shape:', token_type_ids.shape)
     print('attention_mask.shape:', attention_mask.shape)
     print('labels.shape:', labels.shape)
    token_ids:
                                            Ο,
                                                  Ο,
     tensor([[ 101, 2057, 2066, 10733,
                                    102,
                                                        0,
                                                              0,
                                                                    0,
                                                       Ο,
                                                                   0],
                    0, 0,
                                Ο,
                                      Ο,
                                            Ο,
                                                  Ο,
                                                             Ο,
                                                       Ο,
           [ 101, 2002, 2515, 2025, 2066, 18108,
                                                102,
                                                             Ο,
                                                                   0.
                                                                   0]])
               0,
                    Ο,
                        Ο,
                              0,
                                      0,
                                                  0,
                                                       0,
                                                             0,
    token_type_ids:
     attention_mask:
     [1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
    labels:
     tensor([0, 1])
    #-----
    token_ids.shape: torch.Size([2, 20])
    token_type_ids.shape: torch.Size([2, 20])
    attention_mask.shape: torch.Size([2, 20])
    labels.shape: torch.Size([2])
```

4. Network Model

4.1 Hiperparameters

```
[0]: hyperparameters = {
        # description
        'experiment_name': 'BERT_v1',
        #-----
        # training / early stopping
        'max_epochs': 10,
        'patience': 1,
        #-----
        # dataset
        'dataset_class': IMDbDataset,
        'split_train_val': 0.8,
        #-----
        # dataloader
        'batch_size': 8,
        'nworkers': 4,
        # network architecture
        'tokenizer': 'BERT',
        'num_classes': 2,
        'max_length': 512,
        #-----
        # optimizer
        'loss_func': 'CE',
        'opt_name': 'Adam',
        'lr': 1e-5,
        'scheduling_factor': 0.95,
        #-----
        # others
        'manual_seed': 42, # RNG seed
    }
```

4.2 Network model definition

```
def forward(self, input_ids, attention_mask, token_type_ids):
      _, bert_output = self.model(input_ids, attention_mask=attention_mask,_u
→token_type_ids=token_type_ids)
      logits = self.classification_layer(bert_output)
      return logits
  def training_step(self, batch, batch_nb):
      # calculate logits and loss
      input_ids, attention_mask, token_type_ids, label = batch
      y_logits = self(input_ids, attention_mask, token_type_ids)
      loss = F.cross_entropy(y_logits, label)
      #-----
      tqdm_dict = {'gpu_usage': gpu_usage()}
                                             # monitor GPU usage
      tensorboard_logs = {'batch_train_loss': loss} # log batch training loss in_
\rightarrow TensorBoard
      #_____
      #return {'loss': loss, 'log': tensorboard_logs}
      return {'loss': loss,
              'log': tensorboard_logs,
              'progress_bar': tqdm_dict}
  def training_epoch_end(self, outputs):
      # calculate epoch loss based on mini-batch average loss
      avg_loss = torch.stack([x['loss'] for x in outputs]).mean()
      # log training epoch loss in TensorBoard
      tensorboard_logs = {'epoch_train_loss': avg_loss}
      # send 'log' key to the logger (TensorBoard)
      return {'log': tensorboard_logs}
  def validation_step(self, batch, batch_nb):
      # calculate logits and loss
      input_ids, attention_mask, token_type_ids, label = batch
      y_logits = self(input_ids, attention_mask, token_type_ids)
      loss = F.cross_entropy(y_logits, label)
      #----
      # calculate accuracy
      _, y_pred = torch.max(y_logits, dim=1)
      val_acc = accuracy_score(y_pred.cpu(), label.cpu())
      val_acc = torch.tensor(val_acc)
      tqdm_dict = {'gpu_usage': gpu_usage()} # monitor GPU usage
      tensorboard_logs = {'batch_valid_loss': loss} # log batch validation loss in_
\rightarrow TensorBoard
      #_____
      return {'step_val_loss': loss,
              'step_val_acc': val_acc,
              'log': tensorboard_logs,
              'progress_bar': tqdm_dict}
  def validation_epoch_end(self, outputs):
      # calculate validation epoch loss and accuracy
```

```
avg_loss = torch.stack([x['step_val_loss'] for x in outputs]).mean()
    avg_val_acc = torch.stack([x['step_val_acc'] for x in outputs]).mean()
   tensorboard_logs = {'epoch_val_loss': avg_loss, 'avg_val_acc': avg_val_acc}
   tqdm_dict = tensorboard_logs
    #-----
   return {'avg_val_loss': avg_loss,
            'avg_val_acc': avg_val_acc,
           'log': tensorboard_logs,
           'progress_bar': tqdm_dict}
def test_step(self, batch, batch_nb):
   input_ids, attention_mask, token_type_ids, label = batch
   y_logits = self(input_ids, attention_mask, token_type_ids)
    _, y_pred = torch.max(y_logits, dim=1)
   test_acc = accuracy_score(y_pred.cpu(), label.cpu())
   test_acc = torch.tensor(test_acc)
   return {'test_acc': test_acc}
def test_epoch_end(self, outputs):
   avg_test_acc = torch.stack([x['test_acc'] for x in outputs]).mean()
   tensorboard_logs = {'avg_test_acc': avg_test_acc}
   tqdm_dict = tensorboard_logs
    #-----
   return {'avg_test_acc': avg_test_acc,
           'log': tensorboard_logs,
           'progress_bar': tqdm_dict}
def prepare_data(self):
   # load both classes
   X_train_pos = self.load_texts('aclImdb/train/pos')
   X_train_neg = self.load_texts('aclImdb/train/neg')
   X_test_pos = self.load_texts('aclImdb/test/pos')
   X_test_neg = self.load_texts('aclImdb/test/neg')
    # join positive and negative classes
   X_train_raw = X_train_pos + X_train_neg
   self.X_test = X_test_pos + X_test_neg
   y_train_raw = [True] * len(X_train_pos) + [False] * len(X_train_neg)
   self.y_test = [True] * len(X_test_pos) + [False] * len(X_test_neg)
    #-----
    # train/valid split
    c = list(zip(X_train_raw, y_train_raw))
   random.seed(self.hparams.manual_seed)
   random.shuffle(c) # shuffle data before spliting
   X_train_raw, y_train_raw = zip(*c)
   n_train = int(self.hparams.split_train_val * len(X_train_raw))
    # create validation set
   self.X_train = X_train_raw[:n_train]
   self.y_train = y_train_raw[:n_train]
   self.X_valid = X_train_raw[n_train:]
    self.y_valid = y_train_raw[n_train:]
```

```
# create datesets
      self.ds_train = self.DatasetClass(self.X_train, self.y_train, self.tokenizer,_

-max_length=self.hparams.max_length)
      self.ds_valid = self.DatasetClass(self.X_valid, self.y_valid, self.tokenizer,
→max_length=self.hparams.max_length)
      self.ds_test = self.DatasetClass(self.X_test, self.y_test, self.tokenizer,_

-max_length=self.hparams.max_length)
  def train_dataloader(self):
      return DataLoader(
          dataset = self.ds_train,
          batch_size = self.hparams.batch_size,
          drop_last = False,
          shuffle = self.dl_shuffle,
          num_workers=self.hparams.nworkers)
  def val_dataloader(self):
      return DataLoader(
          dataset = self.ds_valid,
          batch_size = self.hparams.batch_size,
          drop_last = False,
          shuffle = False,
          num_workers=self.hparams.nworkers)
  def test_dataloader(self):
      return DataLoader(
          dataset = self.ds_test,
          batch_size = self.hparams.batch_size,
          drop_last = False,
          shuffle = False,
          num_workers=self.hparams.nworkers)
  def configure_optimizers(self):
      optimizer = self.get_optimizer(self.hparams.opt_name, self.hparams.lr)
      scheduler = StepLR(optimizer, 1, self.hparams.scheduling_factor)
      return [optimizer], [scheduler]
  function that returns the loss function associated to a string
  #-----
  parameters:
    loss_func:
      - 'CE': returns the Cross Entropy loss function
      - 'MSE': returns the Mean Squared Error loss function
      - otherwise raise an error
  def get_loss_func(self, loss_func):
      if (loss_func == 'CE'):
          return CrossEntropyLoss()
      elif (loss_func == 'MSE'):
          return MSELoss()
      else:
```

```
raise ValueError(f"Unsupported loss function: {loss_func}")
111
function that returns the optimizer associated to a string
#-----
parameters:
 opt:
   - 'Adam': returns the Adam optimizer
   - 'SGD': returns the Stochastic Gradient Descent optimizer
   - otherwise raise an error
 lr: learning rate
def get_optimizer(self, opt, lr):
   if (opt == 'Adam'):
       #return Adam(self.model.parameters(), lr=self.hparams.lr)
       return Adam( [p for p in self.parameters() if p.requires_grad],
          lr=self.hparams.lr, eps=1e-08)
   elif (opt == 'SGD'):
       return SGD(self.model.parameters(), lr=self.hparams.lr)
   else:
       raise ValueError(f"Unsupported optimizer: {opt}")
function that returns the tokenizer associated to a string
#-----
parameters:
 tokenizer:
   - 'BERT': returns the BERT tokenizer
   - otherwise raise an error
def get_tokenizer(self, tokenizer):
   if (tokenizer == 'BERT'):
       return BertTokenizer.from_pretrained('bert-base-uncased')
   else:
       raise ValueError(f"Unsupported tokenizer: {tokenizer}")
function that reads .txt files and return them as a list
#-----
parameters:
 folder: directory where the .txt files are
# function that reads .txt files and return them as a list
def load_texts(self, folder):
   texts = []
   for path in os.listdir(folder):
       with open(os.path.join(folder, path)) as f:
          texts.append(f.read())
   return texts
```

4.3 Fast dev run (unit test)

```
[18]: # the dictionary of hyperparameters must be converted to a Namespace
      # this way, it is possible to save it alongside with the checkpoint and logger
      model = BertFinetuner(hparams=Namespace(**hyperparameters))
      #-----
      trainer_unit_test = pl.Trainer(
          fast_dev_run = True,
                                        # perform unit test
          profiler = True,
                                         # run profiler
                                        # GPUs
          gpus = 1,
          precision = 32,
                                        # choose precision (32/16 bits)
         )
     INFO:transformers.tokenization_utils:loading file
     https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-vocab.txt
     from cache at /root/.cache/torch/transformers/26bc1ad6c0ac742e9b52263248f6d0f000
     68293b33709fae12320c0e35ccfbbb.542ce4285a40d23a559526243235df47c5f75c197f04f37d1
     a0c124c32c9a084
     INFO:filelock:Lock 140253147788848 acquired on /root/.cache/torch/transformers/4
     dad0251492946e18ac39290fcfe91b89d370fee250efe9521476438fe8ca185.7156163d5fdc189c
     3016 baca 0775 ffce 230789 d7 fa 2a42 ef516483 e4 ca884517.lock
     INFO:transformers.file_utils:https://s3.amazonaws.com/models.huggingface.co/bert
     /bert-base-uncased-config.json not found in cache or force_download set to True,
     downloading to /root/.cache/torch/transformers/tmpa3d_gof9
     HBox(children=(IntProgress(value=0, description='Downloading', max=433, style=ProgressStyle(description
     INFO:transformers.file_utils:storing
     https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
     config.json in cache at /root/.cache/torch/transformers/4dad0251492946e18ac39290
     \verb|fcfe91b89d370fee250efe9521476438fe8ca185.7156163d5fdc189c3016baca0775ffce230789d| \\
     7fa2a42ef516483e4ca884517
     INFO:transformers.file_utils:creating metadata file for /root/.cache/torch/trans
     formers/4dad0251492946e18ac39290fcfe91b89d370fee250efe9521476438fe8ca185.7156163
     d5fdc189c3016baca0775ffce230789d7fa2a42ef516483e4ca884517
     INFO:filelock:Lock 140253147788848 released on /root/.cache/torch/transformers/4
     dad0251492946e18ac39290fcfe91b89d370fee250efe9521476438fe8ca185.7156163d5fdc189c
     3016 baca 0775 ffce 230789 d7 fa 2a42 ef516483 e4 ca884517.lock
     INFO:transformers.configuration_utils:loading configuration file
     https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
     config.json from cache at /root/.cache/torch/transformers/4dad0251492946e18ac392
     90fcfe91b89d370fee250efe9521476438fe8ca185.7156163d5fdc189c3016baca0775ffce23078
     9d7fa2a42ef516483e4ca884517
     INFO:transformers.configuration_utils:Model config BertConfig {
       "_num_labels": 2,
       "architectures": [
         "BertForMaskedLM"
       ],
       "attention_probs_dropout_prob": 0.1,
       "bad_words_ids": null,
       "bos_token_id": null,
```

```
"decoder_start_token_id": null,
"do_sample": false,
"early_stopping": false,
"eos_token_id": null,
"finetuning_task": null,
"hidden_act": "gelu",
"hidden_dropout_prob": 0.1,
"hidden_size": 768,
"id2label": {
  "0": "LABEL_0",
 "1": "LABEL_1"
"initializer_range": 0.02,
"intermediate_size": 3072,
"is_decoder": false,
"is_encoder_decoder": false,
"label2id": {
  "LABEL_0": 0,
  "LABEL 1": 1
},
"layer_norm_eps": 1e-12,
"length_penalty": 1.0,
"max_length": 20,
"max_position_embeddings": 512,
"min_length": 0,
"model_type": "bert",
"no_repeat_ngram_size": 0,
"num_attention_heads": 12,
"num_beams": 1,
"num_hidden_layers": 12,
"num_return_sequences": 1,
"output_attentions": false,
"output_hidden_states": false,
"output_past": true,
"pad_token_id": 0,
"prefix": null,
"pruned_heads": {},
"repetition_penalty": 1.0,
"task_specific_params": null,
"temperature": 1.0,
"top_k": 50,
"top_p": 1.0,
"torchscript": false,
"type_vocab_size": 2,
"use_bfloat16": false,
"vocab_size": 30522
```

}

 $INFO: filelock: Lock 140253302421768 \ acquired \ on \ /root/. cache/torch/transformers/a alef1aede4482d0dbcd4d52baad8ae300e60902e88fcb0bebdec09afd232066.36ca03ab34a1a5d5 fa7bc3d03d55c4fa650fed07220e2eeebc06ce58d0e9a157.lock$

INFO:transformers.file_utils:https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-pytorch_model.bin not found in cache or force_download set to True, downloading to /root/.cache/torch/transformers/tmpOmpnh6ij

HBox(children=(IntProgress(value=0, description='Downloading', max=440473133, style=ProgressStyle(description='Downloading', max=44047313, style=ProgressStyle(description='Downloading',

```
INFO:transformers.file_utils:storing
    https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
    pytorch_model.bin in cache at /root/.cache/torch/transformers/aa1ef1aede4482d0db
    cd4d52baad8ae300e60902e88fcb0bebdec09afd232066.36ca03ab34a1a5d5fa7bc3d03d55c4fa6
    50fed07220e2eeebc06ce58d0e9a157
    INFO:transformers.file_utils:creating metadata file for /root/.cache/torch/trans
    b34a1a5d5fa7bc3d03d55c4fa650fed07220e2eeebc06ce58d0e9a157
    INFO:filelock:Lock 140253302421768 released on /root/.cache/torch/transformers/a
    fa7bc3d03d55c4fa650fed07220e2eeebc06ce58d0e9a157.lock
    INFO:transformers.modeling_utils:loading weights file
    https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
    pytorch_model.bin from cache at /root/.cache/torch/transformers/aa1ef1aede4482d0
    dbcd4d52baad8ae300e60902e88fcb0bebdec09afd232066.36ca03ab34a1a5d5fa7bc3d03d55c4f
    a650fed07220e2eeebc06ce58d0e9a157
    INFO:lightning:Running in fast_dev_run mode: will run a full train, val and test
    loop using a single batch
    INFO:lightning:GPU available: True, used: True
    INFO:lightning:CUDA_VISIBLE_DEVICES: [0]
[19]: model.to(device);
     trainer_unit_test.fit(model)
     trainer_unit_test.test(model)
     del model
    INFO:lightning:
        | Name
                                                        | Type
    Params
       | loss func
                                                        | CrossEntropyLoss | 0
       | model
                                                        | BertModel
                                                                          Ι
    1
    109 M
    2
       | model.embeddings
                                                        | BertEmbeddings
                                                                          | 23
    М
                                                                          | 23
    3
       | model.embeddings.word_embeddings
                                                        | Embedding
    М
                                                        | Embedding
    4
        | model.embeddings.position_embeddings
    393 K
                                                                          | 1
    5
        | model.embeddings.token_type_embeddings
                                                        | Embedding
    K
    6
        | model.embeddings.LayerNorm
                                                        | LayerNorm
                                                                          | 1
    K
    7
        | model.embeddings.dropout
                                                        Dropout
                                                                          10
                                                        | BertEncoder
                                                                          | 85
    8
        | model.encoder
```

| ModuleList

I 85

М

9

| model.encoder.layer

M 10 model.encoder.layer.0 M	BertLayer 7
11 model.encoder.layer.0.attention	BertAttention 2
n 12 model.encoder.layer.O.attention.self M	BertSelfAttention 1
13 model.encoder.layer.0.attention.self.query	Linear
590 K 14 model.encoder.layer.0.attention.self.key	Linear
590 K 15 model.encoder.layer.0.attention.self.value	Linear
590 K 16 model.encoder.layer.0.attention.self.dropout 17 model.encoder.layer.0.attention.output	Dropout
592 K 18 model.encoder.layer.0.attention.output.dense	Linear
590 K 19 model.encoder.layer.O.attention.output.LayerNorm	LayerNorm 1
<pre>K 20 model.encoder.layer.0.attention.output.dropout 21 model.encoder.layer.0.intermediate</pre>	Dropout
M 22 model.encoder.layer.0.intermediate.dense	Linear 2
M 23 model.encoder.layer.0.output	BertOutput 2
M 24 model.encoder.layer.0.output.dense	Linear
M 25 model.encoder.layer.0.output.LayerNorm	LayerNorm 1
<pre>K 26 model.encoder.layer.0.output.dropout 27 model.encoder.layer.1</pre>	Dropout
M 28 model.encoder.layer.1.attention	BertAttention 2
M 29 model.encoder.layer.1.attention.self	BertSelfAttention 1
M 30 model.encoder.layer.1.attention.self.query	Linear
590 K 31 model.encoder.layer.1.attention.self.key 590 K	Linear
32 model.encoder.layer.1.attention.self.value 590 K	Linear
33 model.encoder.layer.1.attention.self.dropout	Dropout
592 K	- -
35 model.encoder.layer.1.attention.output.dense 590 K	Linear
36 model.encoder.layer.1.attention.output.LayerNorm K	LayerNorm 1
<pre>37 model.encoder.layer.1.attention.output.dropout 38 model.encoder.layer.1.intermediate M</pre>	Dropout

39	1	model encoder layer 1 intermediate dence	1	Linear	ı	2
39 M	1	model.encoder.layer.1.intermediate.dense	1	Linear	ı	2
40 M	1	model.encoder.layer.1.output	I	BertOutput	I	2
41 M	١	model.encoder.layer.1.output.dense	I	Linear	I	2
42 K	I	model.encoder.layer.1.output.LayerNorm	I	LayerNorm	I	1
43	1	model.encoder.layer.1.output.dropout	1	Dropout	1	0
44 M	I	model.encoder.layer.2	I	BertLayer	I	7
45 M	1	model.encoder.layer.2.attention	I	BertAttention	I	2
46 M	١	model.encoder.layer.2.attention.self	I	BertSelfAttention	I	1
47 590		model.encoder.layer.2.attention.self.query	I	Linear	I	
48 590		model.encoder.layer.2.attention.self.key	I	Linear	I	
49 590		model.encoder.layer.2.attention.self.value	I	Linear	I	
50		model.encoder.layer.2.attention.self.dropout	1	Dropout	1	0
51	1	model.encoder.layer.2.attention.output	1	BertSelfOutput	1	
592	K					
52 590		model.encoder.layer.2.attention.output.dense	I	Linear	1	
53 K		model.encoder.layer.2.attention.output.LayerNorm	I	LayerNorm	I	1
54	1	model.encoder.layer.2.attention.output.dropout	-	Dropout	-	0
55		model.encoder.layer.2.intermediate	-	${\tt BertIntermediate}$		2
M						
56 M	1	model.encoder.layer.2.intermediate.dense	ı	Linear	I	2
57 M		model.encoder.layer.2.output	I	BertOutput	I	2
58 M	١	model.encoder.layer.2.output.dense	I	Linear	I	2
59 K	1	model.encoder.layer.2.output.LayerNorm	I	LayerNorm	1	1
60	1	model.encoder.layer.2.output.dropout	-	Dropout	-	0
61	1	model.encoder.layer.3	-	BertLayer	-	7
M						
62 M	1	model.encoder.layer.3.attention	I	BertAttention	I	2
63 M	١	model.encoder.layer.3.attention.self	I	BertSelfAttention	1	1
64 590		model.encoder.layer.3.attention.self.query	I	Linear	I	
65 590		model.encoder.layer.3.attention.self.key	١	Linear	I	
66 590		model.encoder.layer.3.attention.self.value	١	Linear	I	
67	1	model.encoder.layer.3.attention.self.dropout	1	Dropout	1	0
68		model.encoder.layer.3.attention.output		BertSelfOutput	I	

500 W	
592 K 69 model.encoder.layer.3.attention.output.dense	Linear
590 K 70 model.encoder.layer.3.attention.output.LayerNorm	LayerNorm 1
K	
71 model.encoder.layer.3.attention.output.dropout	Dropout 0
72 model.encoder.layer.3.intermediate	BertIntermediate 2
73 model.encoder.layer.3.intermediate.dense	Linear 2
M 74 model.encoder.layer.3.output	BertOutput 2
M	1.7.
75 model.encoder.layer.3.output.dense	Linear 2
76 model.encoder.layer.3.output.LayerNorm K	LayerNorm 1
77 model.encoder.layer.3.output.dropout	Dropout 0
78 model.encoder.layer.4	BertLayer 7
M	
79 model.encoder.layer.4.attention	BertAttention 2
80 model.encoder.layer.4.attention.self	BertSelfAttention 1
81 model.encoder.layer.4.attention.self.query	Linear
590 K 82 model.encoder.layer.4.attention.self.key	Linear
82 model.encoder.layer.4.attention.self.key 590 K	Linear
83 model.encoder.layer.4.attention.self.value	Linear
590 K	
84 model.encoder.layer.4.attention.self.dropout	Dropout 0
85 model.encoder.layer.4.attention.output	BertSelfOutput
592 K	
86 model.encoder.layer.4.attention.output.dense	Linear
590 K 87 model.encoder.layer.4.attention.output.LayerNorm	LayerNorm 1
K	LayerNorm 1
88 model.encoder.layer.4.attention.output.dropout	Dropout 0
89 model.encoder.layer.4.intermediate	BertIntermediate 2
M 90 model.encoder.layer.4.intermediate.dense	Linear 2
90 model.encoder.layer.4.intermediate.dense	Linear 2
91 model.encoder.layer.4.output	BertOutput 2
92 model.encoder.layer.4.output.dense	Linear 2
M	
93 model.encoder.layer.4.output.LayerNorm	LayerNorm 1
<pre>K 94 model.encoder.layer.4.output.dropout</pre>	Dropout
94 model.encoder.layer.4.output.dropout 95 model.encoder.layer.5	Dropout
M M	I ner organia I I
96 model.encoder.layer.5.attention	BertAttention 2
M	
97 model.encoder.layer.5.attention.self	BertSelfAttention 1
M	

98 model.encoder.layer.5.attention.self.query 590 K	I	Linear	١	
99 model.encoder.layer.5.attention.self.key 590 K	I	Linear	١	
100 model.encoder.layer.5.attention.self.value 590 K	I	Linear	١	
101 model.encoder.layer.5.attention.self.dropout 102 model.encoder.layer.5.attention.output		Dropout BertSelfOutput	1	0
592 K 103 model.encoder.layer.5.attention.output.dense 590 K	I	Linear	I	
104 model.encoder.layer.5.attention.output.LayerNorm	I	LayerNorm	١	1
105 model.encoder.layer.5.attention.output.dropout	- 1	Dropout	ı	0
106 model.encoder.layer.5.intermediate		BertIntermediate	i	2
M	'	Del diliterimediate	'	_
107 model.encoder.layer.5.intermediate.dense	I	Linear	١	2
108 model.encoder.layer.5.output	I	BertOutput	١	2
109 model.encoder.layer.5.output.dense	I	Linear	١	2
110 model.encoder.layer.5.output.LayerNorm	1	LayerNorm	١	1
111 model.encoder.layer.5.output.dropout	- 1	Dropout		0
112 model.encoder.layer.6	I	BertLayer	1	7
113 model.encoder.layer.6.attention	I	BertAttention	1	2
114 model.encoder.layer.6.attention.self		BertSelfAttention	1	1
115 model.encoder.layer.6.attention.self.query 590 K		Linear		
116 model.encoder.layer.6.attention.self.key 590 K	·	Linear		
117 model.encoder.layer.6.attention.self.value 590 K		Linear	1	
118 model.encoder.layer.6.attention.self.dropout		Dropout		0
119 model.encoder.layer.6.attention.output	ı	BertSelfOutput	ı	
592 K				
120 model.encoder.layer.6.attention.output.dense	ı	Linear	ı	
590 K				
121 model.encoder.layer.6.attention.output.LayerNorm K		LayerNorm		1
122 model.encoder.layer.6.attention.output.dropout		Dropout		0
123 model.encoder.layer.6.intermediate	I	BertIntermediate	ı	2
124 model.encoder.layer.6.intermediate.dense	I	Linear	١	2
125 model.encoder.layer.6.output M	I	BertOutput	١	2
126 model.encoder.layer.6.output.dense	1	Linear	١	2
127 model.encoder.layer.6.output.LayerNorm	I	LayerNorm	١	1

7.7						
		model.encoder.layer.6.output.dropout		Dropout	I	0
129 M)	model.encoder.layer.7	I	BertLayer	١	7
130 M)	model.encoder.layer.7.attention	١	BertAttention	I	2
	. 1	model.encoder.layer.7.attention.self	I	${\tt BertSelfAttention}$	١	1
132		model.encoder.layer.7.attention.self.query	I	Linear	١	
	3	model.encoder.layer.7.attention.self.key	ı	Linear	ı	
590 134		model.encoder.layer.7.attention.self.value	ı	Linear	ı	
590						
		model.encoder.layer.7.attention.self.dropout		Dropout		0
		model.encoder.layer.7.attention.output		BertSelfOutput		
592 137		model.encoder.layer.7.attention.output.dense	ı	Linear	ı	
590			'	Linear	'	
		model.encoder.layer.7.attention.output.LayerNorm	1	LayerNorm	Ι	1
K				·		
139	1	model.encoder.layer.7.attention.output.dropout	-	Dropout	-	0
140		model.encoder.layer.7.intermediate	-	${\tt BertIntermediate}$		2
M						
141	.	model.encoder.layer.7.intermediate.dense	I	Linear	1	2
М						
	?	model.encoder.layer.7.output	ı	BertOutput	ı	2
M				T ·		_
143 M)	model.encoder.layer.7.output.dense	١	Linear	١	2
	. 1	model.encoder.layer.7.output.LayerNorm	ı	LayerNorm	ı	1
K		model.encodel.layel./.output.LayelNolm	'	LayerNorm	'	_
	;	model.encoder.layer.7.output.dropout	ı	Dropout	ı	0
		model.encoder.layer.8		BertLayer		7
М	•		•	,	•	
147	'	model.encoder.layer.8.attention	1	BertAttention	1	2
М						
148	3	model.encoder.layer.8.attention.self	I	${\tt BertSelfAttention}$		1
М						
		model.encoder.layer.8.attention.self.query	ı	Linear	ı	
590				T :		
590		model.encoder.layer.8.attention.self.key	١	Linear	ı	
		model.encoder.layer.8.attention.self.value	ı	Linear	ı	
590		model.encodel.layer.o.accention.serr.varue	'	Lincai	'	
		model.encoder.layer.8.attention.self.dropout	ı	Dropout	ı	0
		model.encoder.layer.8.attention.output		BertSelfOutput	i	-
592		,	•		•	
		model.encoder.layer.8.attention.output.dense	ı	Linear	1	
590		•				
155	5	model.encoder.layer.8.attention.output.LayerNorm	-	LayerNorm	1	1
K						
		model.encoder.layer.8.attention.output.dropout		Dropout		0
157	'	model.encoder.layer.8.intermediate	I	BertIntermediate	١	2

M 158 model.encoder.layer.8.intermediate.dense	Linear	2
M 159 model.encoder.layer.8.output M	BertOutput	2
160 model.encoder.layer.8.output.dense	Linear	2
161 model.encoder.layer.8.output.LayerNorm	LayerNorm	1
162 model.encoder.layer.8.output.dropout 163 model.encoder.layer.9	Dropout BertLayer	0 7
M	v	
164 model.encoder.layer.9.attention		2
165 model.encoder.layer.9.attention.self	BertSelfAttention	1
166 model.encoder.layer.9.attention.self.query 590 K	Linear	
167 model.encoder.layer.9.attention.self.key 590 K	Linear	
168 model.encoder.layer.9.attention.self.value 590 K	Linear	
169 model.encoder.layer.9.attention.self.dropout	Dropout	0
170 model.encoder.layer.9.attention.output 592 K	BertSelfOutput	
171 model.encoder.layer.9.attention.output.dense 590 K	Linear	
172 model.encoder.layer.9.attention.output.LayerNorm K	LayerNorm	1
173 model.encoder.layer.9.attention.output.dropout	Dropout	0
174 model.encoder.layer.9.intermediate	BertIntermediate	2
175 model.encoder.layer.9.intermediate.dense	Linear	2
176 model.encoder.layer.9.output M	BertOutput	2
177 model.encoder.layer.9.output.dense	Linear	2
178 model.encoder.layer.9.output.LayerNorm K	LayerNorm	1
179 model.encoder.layer.9.output.dropout	· •	0
180 model.encoder.layer.10	BertLayer	7
181 model.encoder.layer.10.attention	BertAttention	2
182 model.encoder.layer.10.attention.self	BertSelfAttention	1
183 model.encoder.layer.10.attention.self.query 590 K	Linear	
184 model.encoder.layer.10.attention.self.key 590 K	Linear	
185 model.encoder.layer.10.attention.self.value 590 K	Linear	
186 model.encoder.layer.10.attention.self.dropout	Dropout	0

187 592		model.encoder.layer.10.attention.output	I	BertSelfOutput	I	
188 590		model.encoder.layer.10.attention.output.dense	I	Linear	١	
189 K	1	<pre>model.encoder.layer.10.attention.output.LayerNorm</pre>	I	LayerNorm	١	1
		model.encoder.layer.10.attention.output.dropout model.encoder.layer.10.intermediate		Dropout BertIntermediate	1	0
	I	model.encoder.layer.10.intermediate.dense	I	Linear	I	2
	I	model.encoder.layer.10.output	I	BertOutput	1	2
	I	model.encoder.layer.10.output.dense	I	Linear	I	2
195 K	1	model.encoder.layer.10.output.LayerNorm	I	LayerNorm	I	1
		<pre>model.encoder.layer.10.output.dropout model.encoder.layer.11</pre>		Dropout BertLayer		0 7
	I	model.encoder.layer.11.attention	ı	BertAttention	I	2
M 199 M	I	model.encoder.layer.11.attention.self	I	BertSelfAttention	I	1
		model.encoder.layer.11.attention.self.query	I	Linear	I	
	١	model.encoder.layer.11.attention.self.key	I	Linear	I	
202 590		model.encoder.layer.11.attention.self.value	I	Linear	I	
203	1	model.encoder.layer.11.attention.self.dropout	1	Dropout	1	0
204	1	model.encoder.layer.11.attention.output	1	BertSelfOutput	1	
592		•		•		
205 590		model.encoder.layer.11.attention.output.dense	I	Linear	I	
206 K		model.encoder.layer.11.attention.output.LayerNorm	1	LayerNorm		1
207		model.encoder.layer.11.attention.output.dropout		Dropout		0
208 M	1	model.encoder.layer.11.intermediate	I	BertIntermediate		2
M		model.encoder.layer.11.intermediate.dense		Linear		2
M		model.encoder.layer.11.output		BertOutput		2
M		model.encoder.layer.11.output.dense		Linear		2
K		model.encoder.layer.11.output.LayerNorm		LayerNorm		1
		model.encoder.layer.11.output.dropout		Dropout	1	0
		model.pooler	١	BertPooler	1	
590						
215 590		model.pooler.dense	I	Linear	I	
		model.pooler.activation	I	Tanh	I	0

```
HBox(children=(IntProgress(value=1, bar_style='info', description='Training', layout=Layout(flex='2'),
HBox(children=(IntProgress(value=1, bar_style='info', description='Validating', layout=Layout(flex='2')
INFO:lightning:
Profiler Report
Action

on_train_start | 0.03086
on_epoch_start | 0.0025166
get_train_batch | 0.35123

batch_start | 0.00010312
| 0.1594
                  | Mean duration (s) | Total time (s)
______
                                         0.03086
                                         0.0025166
                                         0.35123
                                        0.00010312
model_forward | 0.1594

model_backward | 0.36685

on_after_backward | 3.522e-06

ontimizer_step | 0.065248
                                         0.1594
                                          0.36685
                                       | 3.522e-06
optimizer_step
                    0.065248
                                         0.065248
                  on_batch_end
on_epoch_end
on_train_end
HBox(children=(IntProgress(value=1, bar_style='info', description='Testing', layout=Layout(flex='2'), m
______
TEST RESULTS
```

| Linear | 1

4.4 Batch overfitting

{'avg_test_acc': tensor(0.2500, dtype=torch.float64)}

217 | classification_layer

```
INFO:transformers.tokenization_utils:loading file
https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-vocab.txt
from cache at /root/.cache/torch/transformers/26bc1ad6c0ac742e9b52263248f6d0f000
68293b33709fae12320c0e35ccfbbb.542ce4285a40d23a559526243235df47c5f75c197f04f37d1
a0c124c32c9a084
INFO:transformers.configuration_utils:loading configuration file
https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
config.json from cache at /root/.cache/torch/transformers/4dad0251492946e18ac392
90fcfe91b89d370fee250efe9521476438fe8ca185.7156163d5fdc189c3016baca0775ffce23078
9d7fa2a42ef516483e4ca884517
INFO:transformers.configuration_utils:Model config BertConfig {
  "_num_labels": 2,
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bad_words_ids": null,
  "bos_token_id": null,
  "decoder_start_token_id": null,
  "do_sample": false,
  "early_stopping": false,
  "eos_token_id": null,
  "finetuning_task": null,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "id2label": {
   "0": "LABEL_0",
   "1": "LABEL_1"
  },
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "is_decoder": false,
  "is_encoder_decoder": false,
  "label2id": {
    "LABEL_0": 0,
    "LABEL_1": 1
  "layer_norm_eps": 1e-12,
  "length_penalty": 1.0,
  "max_length": 20,
  "max_position_embeddings": 512,
  "min_length": 0,
  "model_type": "bert",
  "no_repeat_ngram_size": 0,
  "num_attention_heads": 12,
  "num_beams": 1,
  "num_hidden_layers": 12,
  "num_return_sequences": 1,
  "output_attentions": false,
  "output_hidden_states": false,
  "output_past": true,
  "pad_token_id": 0,
  "prefix": null,
```

```
"pruned_heads": {},
       "repetition_penalty": 1.0,
       "task_specific_params": null,
       "temperature": 1.0,
       "top_k": 50,
       "top_p": 1.0,
       "torchscript": false,
       "type_vocab_size": 2,
       "use_bfloat16": false,
       "vocab_size": 30522
     }
     INFO:transformers.modeling_utils:loading weights file
     https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
     pytorch_model.bin from cache at /root/.cache/torch/transformers/aa1ef1aede4482d0
     a650fed07220e2eeebc06ce58d0e9a157
     INFO:lightning:GPU available: True, used: True
     INFO:lightning:CUDA_VISIBLE_DEVICES: [0]
[21]: model.to(device)
     trainer_batch_overfit.fit(model)
     trainer_batch_overfit.test(model)
      #del model
     INFO:lightning:
                                                           | Type
                                                                              1
        | Name
     Params
       | loss_func
                                                           | CrossEntropyLoss | 0
                                                           | BertModel
     1
        | model
                                                                              1
     109 M
     2
        | model.embeddings
                                                           | BertEmbeddings
                                                                              | 23
     М
     3
         | model.embeddings.word_embeddings
                                                           | Embedding
                                                                              | 23
     М
     4
         | model.embeddings.position_embeddings
                                                           | Embedding
     393 K
         | model.embeddings.token_type_embeddings
                                                           | Embedding
                                                                              | 1
     5
     K
     6
         | model.embeddings.LayerNorm
                                                           | LayerNorm
                                                                              | 1
     K
         | model.embeddings.dropout
     7
                                                           | Dropout
                                                                              1 0
     8
         | model.encoder
                                                           | BertEncoder
                                                                              | 85
     М
         | model.encoder.layer
                                                           | ModuleList
                                                                              | 85
     9
     М
     10
        | model.encoder.layer.0
                                                           | BertLayer
                                                                              | 7
     М
                                                           | BertAttention
     11
        | model.encoder.layer.O.attention
                                                                              1 2
     М
        | model.encoder.layer.0.attention.self
                                                           | BertSelfAttention | 1
     12
     М
```

```
13 | model.encoder.layer.O.attention.self.query
                                                          | Linear
590 K
   | model.encoder.layer.O.attention.self.key
14
                                                          | Linear
590 K
     model.encoder.layer.0.attention.self.value
15
                                                          | Linear
590 K
    | model.encoder.layer.O.attention.self.dropout
                                                          | Dropout
                                                                              10
     model.encoder.layer.0.attention.output
                                                          | BertSelfOutput
17
592 K
    | model.encoder.layer.O.attention.output.dense
18
                                                          Linear
590 K
     model.encoder.layer.O.attention.output.LayerNorm
19
                                                         | LayerNorm
                                                                              | 1
K
    | model.encoder.layer.O.attention.output.dropout
                                                          | Dropout
                                                                              1 0
20
21
    | model.encoder.layer.O.intermediate
                                                          | BertIntermediate
                                                                              1 2
М
22
    | model.encoder.layer.O.intermediate.dense
                                                          Linear
                                                                              1 2
М
23
    | model.encoder.layer.0.output
                                                          | BertOutput
                                                                              1 2
М
24
    | model.encoder.layer.O.output.dense
                                                          | Linear
                                                                              1 2
М
25
    | model.encoder.layer.O.output.LayerNorm
                                                          | LayerNorm
                                                                              | 1
26
    | model.encoder.layer.O.output.dropout
                                                          | Dropout
                                                                              1 0
27
    | model.encoder.layer.1
                                                          | BertLayer
                                                                              17
М
28
   | model.encoder.layer.1.attention
                                                          | BertAttention
М
   | model.encoder.layer.1.attention.self
                                                          | BertSelfAttention | 1
29
М
30
   | model.encoder.layer.1.attention.self.query
                                                          | Linear
590 K
   | model.encoder.layer.1.attention.self.key
                                                          | Linear
31
590 K
   | model.encoder.layer.1.attention.self.value
32
                                                          | Linear
590 K
33
    | model.encoder.layer.1.attention.self.dropout
                                                          | Dropout
                                                                              10
     model.encoder.layer.1.attention.output
                                                          | BertSelfOutput
34
592 K
    | model.encoder.layer.1.attention.output.dense
                                                          | Linear
590 K
    | model.encoder.layer.1.attention.output.LayerNorm
                                                          | LayerNorm
36
K
    | model.encoder.layer.1.attention.output.dropout
                                                          | Dropout
                                                                              1 0
37
                                                          | BertIntermediate
    | model.encoder.layer.1.intermediate
38
М
    | model.encoder.layer.1.intermediate.dense
                                                          | Linear
                                                                              1 2
39
М
    | model.encoder.layer.1.output
                                                          | BertOutput
                                                                              1 2
40
М
   | model.encoder.layer.1.output.dense
                                                          | Linear
                                                                              | 2
41
М
   | model.encoder.layer.1.output.LayerNorm
                                                          | LayerNorm
                                                                              1 1
42
```

K						
43	I	model.encoder.layer.1.output.dropout	I	Dropout	1	0
44 M		model.encoder.layer.2		BertLayer	I	7
M 45 M	I	model.encoder.layer.2.attention	I	BertAttention	I	2
46 M	I	model.encoder.layer.2.attention.self	I	BertSelfAttention	I	1
47 590		model.encoder.layer.2.attention.self.query	I	Linear	I	
48 590	-	model.encoder.layer.2.attention.self.key	I	Linear	I	
49 590	-	model.encoder.layer.2.attention.self.value	I	Linear	I	
50		model.encoder.layer.2.attention.self.dropout	ı	Dropout	ı	0
51	i	<u> </u>		BertSelfOutput	i	Ŭ
592	K	·		•		
52 590		model.encoder.layer.2.attention.output.dense	I	Linear	١	
53		model.encoder.layer.2.attention.output.LayerNorm	ı	LayerNorm	ı	1
K	•		•		•	_
54	-	model.encoder.layer.2.attention.output.dropout	-	Dropout	-	0
55	I	model.encoder.layer.2.intermediate	I	BertIntermediate		2
M 56	1	model.encoder.layer.2.intermediate.dense		Linear	1	2
M	'	model.encodel.layer.z.intermediate.dense		Tillegi	ı	2
57	1	model.encoder.layer.2.output	-	BertOutput	1	2
M						
58 M	١	model.encoder.layer.2.output.dense	ı	Linear	١	2
59	I	model.encoder.layer.2.output.LayerNorm	I	LayerNorm	I	1
K 60	ı	model.encoder.layer.2.output.dropout	ı	Dropout	ı	0
61		model.encoder.layer.3		BertLayer	i	7
M	-	J	-	v		
62 M		model.encoder.layer.3.attention	I	BertAttention	I	2
M 63	I	model.encoder.layer.3.attention.self	I	BertSelfAttention	١	1
M				Times		
64 590		model.encoder.layer.3.attention.self.query	ı	Linear	ı	
65		model.encoder.layer.3.attention.self.key	ı	Linear	١	
590		J	-			
66		model.encoder.layer.3.attention.self.value	I	Linear		
590				ъ.		^
67 68	l I	model.encoder.layer.3.attention.self.dropout model.encoder.layer.3.attention.output		Dropout BertSelfOutput	1	0
592	K	model.encodel.layel.S.actention.output	'	Der cherroachac	1	
69		model.encoder.layer.3.attention.output.dense	١	Linear	١	
590						
70 K	١	model.encoder.layer.3.attention.output.LayerNorm	I	LayerNorm	I	1
к 71	ı	model.encoder.layer.3.attention.output.dropout	ı	Dropout	ı	0
72		model.encoder.layer.3.intermediate	i	BertIntermediate	i	2
		•				

```
М
73
    | model.encoder.layer.3.intermediate.dense
                                                          | Linear
                                                                              1 2
М
    | model.encoder.layer.3.output
                                                          | BertOutput
                                                                              1 2
74
М
75
    | model.encoder.layer.3.output.dense
                                                          | Linear
                                                                              | 2
М
76
    | model.encoder.layer.3.output.LayerNorm
                                                          | LayerNorm
                                                                              1 1
K
    | model.encoder.layer.3.output.dropout
                                                          | Dropout
                                                                              1 0
77
78
    | model.encoder.layer.4
                                                          | BertLayer
М
    | model.encoder.layer.4.attention
                                                          | BertAttention
79
М
    | model.encoder.layer.4.attention.self
80
                                                          | BertSelfAttention | 1
М
    | model.encoder.layer.4.attention.self.query
                                                          | Linear
81
590 K
     model.encoder.layer.4.attention.self.key
                                                          | Linear
82
590 K
83
    | model.encoder.layer.4.attention.self.value
                                                          | Linear
590 K
                                                                              10
    | model.encoder.layer.4.attention.self.dropout
                                                          | Dropout
84
     model.encoder.layer.4.attention.output
                                                          | BertSelfOutput
85
592 K
86
    | model.encoder.layer.4.attention.output.dense
                                                          | Linear
590 K
    | model.encoder.layer.4.attention.output.LayerNorm
                                                          | LayerNorm
87
K
88
    | model.encoder.layer.4.attention.output.dropout
                                                          | Dropout
                                                          | BertIntermediate
89
    | model.encoder.layer.4.intermediate
М
    | model.encoder.layer.4.intermediate.dense
                                                                              1 2
90
                                                          Linear
М
91
    | model.encoder.layer.4.output
                                                          | BertOutput
                                                                              | 2
М
92
    | model.encoder.layer.4.output.dense
                                                          | Linear
                                                                              1 2
М
93
    | model.encoder.layer.4.output.LayerNorm
                                                          | LayerNorm
K
    | model.encoder.layer.4.output.dropout
                                                          | Dropout
                                                                              10
95
    | model.encoder.layer.5
                                                          | BertLayer
                                                                              | 7
М
    | model.encoder.layer.5.attention
                                                          | BertAttention
                                                                              1 2
96
М
                                                          | BertSelfAttention | 1
    | model.encoder.layer.5.attention.self
97
М
   | model.encoder.layer.5.attention.self.query
                                                          Linear
98
590 K
    | model.encoder.layer.5.attention.self.key
                                                          | Linear
99
590 K
100 | model.encoder.layer.5.attention.self.value
                                                          | Linear
590 K
101 | model.encoder.layer.5.attention.self.dropout
                                                          | Dropout
                                                                              10
```

102 592		model.encoder.layer.5.attention.output	I	BertSelfOutput	1	
103 590		model.encoder.layer.5.attention.output.dense	I	Linear	I	
104 K	I	model.encoder.layer.5.attention.output.LayerNorm	I	LayerNorm	I	1
105		model.encoder.layer.5.attention.output.dropout	1	Dropout		0
106 M		model.encoder.layer.5.intermediate	I	BertIntermediate		2
107 M	1	model.encoder.layer.5.intermediate.dense	I	Linear		2
108 M		model.encoder.layer.5.output	I	BertOutput		2
109 M	1	model.encoder.layer.5.output.dense	I	Linear		2
110 K	1	model.encoder.layer.5.output.LayerNorm	I	LayerNorm		1
		model.encoder.layer.5.output.dropout		Dropout	1	0
М		model.encoder.layer.6		BertLayer	1	7
М		model.encoder.layer.6.attention		BertAttention	·	2
М		model.encoder.layer.6.attention.self		BertSelfAttention	1	1
590	K	model.encoder.layer.6.attention.self.query		Linear	1	
590	K	model.encoder.layer.6.attention.self.key		Linear	1	
590	K	model.encoder.layer.6.attention.self.value		Linear		
		model.encoder.layer.6.attention.self.dropout		Dropout	!	0
592	K	model.encoder.layer.6.attention.output		BertSelfOutput	1	
590		model.encoder.layer.6.attention.output.dense	I	Linear	ı	
121 K	I	model.encoder.layer.6.attention.output.LayerNorm	I	LayerNorm	I	1
122	1	model.encoder.layer.6.attention.output.dropout		Dropout	1	0
123 M	1	model.encoder.layer.6.intermediate	I	BertIntermediate	1	2
124 M	1	model.encoder.layer.6.intermediate.dense	I	Linear		2
125 M	1	model.encoder.layer.6.output	I	BertOutput		2
М		model.encoder.layer.6.output.dense		Linear		2
K		model.encoder.layer.6.output.LayerNorm		LayerNorm	1	1
		model.encoder.layer.6.output.dropout		Dropout		0
М		model.encoder.layer.7		BertLayer		7
М		model.encoder.layer.7.attention		BertAttention		2
131	ı	model.encoder.layer.7.attention.self	1	BertSelfAttention		1

M			· ·		
132 590 K	model.encoder.layer.7.attention.self.query	ı	Linear	ı	
133 590 K	model.encoder.layer.7.attention.self.key	١	Linear	١	
	model.encoder.layer.7.attention.self.value	١	Linear	١	
135	model.encoder.layer.7.attention.self.dropout		Dropout	١	0
136 592 K	model.encoder.layer.7.attention.output	١	BertSelfOutput	١	
	model.encoder.layer.7.attention.output.dense	ı	Linear	ı	
590 K	, i				
138 K	model.encoder.layer.7.attention.output.LayerNorm		LayerNorm	I	1
	model.encoder.layer.7.attention.output.dropout		Dropout		0
140 M	model.encoder.layer.7.intermediate	١	BertIntermediate	١	2
141 M	model.encoder.layer.7.intermediate.dense	١	Linear	I	2
142 M	model.encoder.layer.7.output	I	BertOutput	I	2
	model.encoder.layer.7.output.dense	١	Linear	I	2
144	model.encoder.layer.7.output.LayerNorm	١	LayerNorm	١	1
K 145 l	model.encoder.layer.7.output.dropout	ı	Dropout	ı	0
	model.encoder.layer.8		BertLayer		7
M	·		·		
147 M	model.encoder.layer.8.attention	I	BertAttention	١	2
148 M	model.encoder.layer.8.attention.self	١	BertSelfAttention	I	1
149 590 K	model.encoder.layer.8.attention.self.query	I	Linear	١	
	model.encoder.layer.8.attention.self.key	I	Linear	١	
151	model.encoder.layer.8.attention.self.value	I	Linear	١	
590 K	model.encoder.layer.8.attention.self.dropout	ı	Dropout	ı	0
	model.encoder.layer.8.attention.output		BertSelfOutput	i	
592 K					
	model.encoder.layer.8.attention.output.dense	ı	Linear	ı	
590 K 155	model.encoder.layer.8.attention.output.LayerNorm	ı	LayerNorm	ı	1
K	1 3		J		
	model.encoder.layer.8.attention.output.dropout		Dropout		0
157 M	model.encoder.layer.8.intermediate	ı	BertIntermediate	ı	2
158	model.encoder.layer.8.intermediate.dense	١	Linear	١	2
	model.encoder.layer.8.output	١	BertOutput	١	2
M 160 M	model.encoder.layer.8.output.dense	I	Linear	I	2
1.1					

161 K	I	model.encoder.layer.8.output.LayerNorm	I	LayerNorm	I	1
		<pre>model.encoder.layer.8.output.dropout model.encoder.layer.9</pre>		Dropout BertLayer		0 7
	١	model.encoder.layer.9.attention	١	BertAttention	١	2
	I	model.encoder.layer.9.attention.self	I	BertSelfAttention	I	1
		model.encoder.layer.9.attention.self.query	I	Linear	I	
	I	model.encoder.layer.9.attention.self.key	1	Linear	١	
	I	model.encoder.layer.9.attention.self.value	١	Linear	I	
170	I	<pre>model.encoder.layer.9.attention.self.dropout model.encoder.layer.9.attention.output</pre>		Dropout BertSelfOutput	 	0
592 171 590	1	model.encoder.layer.9.attention.output.dense	I	Linear	I	
		model.encoder.layer.9.attention.output.LayerNorm	I	LayerNorm	I	1
	ı	model.encoder.layer.9.attention.output.dropout	ı	Dropout	ı	0
		model.encoder.layer.9.intermediate		BertIntermediate	Ì	2
175 M	١	model.encoder.layer.9.intermediate.dense	1	Linear		2
176 M		model.encoder.layer.9.output		BertOutput		2
177 M	١	model.encoder.layer.9.output.dense	1	Linear	1	2
178 K	1	model.encoder.layer.9.output.LayerNorm		LayerNorm	1	1
		model.encoder.layer.9.output.dropout		Dropout		0
M		model.encoder.layer.10		BertLayer	1	7
M		model.encoder.layer.10.attention	•	BertAttention	-	2
M		model.encoder.layer.10.attention.self		BertSelfAttention		1
590	K	model.encoder.layer.10.attention.self.query		Linear		
590	K	model.encoder.layer.10.attention.self.key		Linear		
590	K	model.encoder.layer.10.attention.self.value		Linear	1	
		model.encoder.layer.10.attention.self.dropout		Dropout	!	0
		model.encoder.layer.10.attention.output	1	BertSelfOutput		
592		model encoder layer 10 sttenties settlet describe	ı	linoar	ı	
188 590		model.encoder.layer.10.attention.output.dense	1	Linear	ı	
		model.encoder.layer.10.attention.output.LayerNorm	I	LayerNorm	I	1
	I	model.encoder.layer.10.attention.output.dropout		Dropout	I	0

```
191 | model.encoder.layer.10.intermediate
                                                         | BertIntermediate | 2
М
192 | model.encoder.layer.10.intermediate.dense
                                                         | Linear
                                                                              1 2
193 | model.encoder.layer.10.output
                                                         | BertOutput
                                                                              | 2
194 | model.encoder.layer.10.output.dense
                                                         | Linear
                                                                              1 2
195 | model.encoder.layer.10.output.LayerNorm
                                                         | LayerNorm
196 | model.encoder.layer.10.output.dropout
                                                         | Dropout
                                                                              1 0
197 | model.encoder.layer.11
                                                         | BertLayer
                                                                              | 7
198 | model.encoder.layer.11.attention
                                                         | BertAttention
                                                                              1 2
199 | model.encoder.layer.11.attention.self
                                                         | BertSelfAttention | 1
200 | model.encoder.layer.11.attention.self.query
                                                         | Linear
590 K
201 | model.encoder.layer.11.attention.self.key
                                                         | Linear
590 K
202 | model.encoder.layer.11.attention.self.value
                                                         | Linear
590 K
203 | model.encoder.layer.11.attention.self.dropout
                                                         | Dropout
                                                                              10
204 | model.encoder.layer.11.attention.output
                                                         | BertSelfOutput
205 | model.encoder.layer.11.attention.output.dense
                                                         | Linear
590 K
206 | model.encoder.layer.11.attention.output.LayerNorm | LayerNorm
                                                                              | 1
207 | model.encoder.layer.11.attention.output.dropout
                                                         | Dropout
                                                                              1 0
208 | model.encoder.layer.11.intermediate
                                                         | BertIntermediate
209 | model.encoder.layer.11.intermediate.dense
                                                         | Linear
                                                                              1 2
210 | model.encoder.layer.11.output
                                                         | BertOutput
                                                                              1 2
211 | model.encoder.layer.11.output.dense
                                                         | Linear
                                                                              1 2
212 | model.encoder.layer.11.output.LayerNorm
                                                         | LayerNorm
                                                                              | 1
213 | model.encoder.layer.11.output.dropout
                                                         Dropout
                                                                              1 0
214 | model.pooler
                                                         | BertPooler
590 K
215 | model.pooler.dense
                                                         | Linear
590 K
216 | model.pooler.activation
                                                         | Tanh
                                                                              1 0
217 | classification_layer
                                                         | Linear
                                                                              | 1
```

HBox(children=(IntProgress(value=1, bar_style='info', description='Validation sanity check', layout=Layout=

```
HBox(children=(IntProgress(value=1, bar_style='info', description='Training', layout=Layout(flex='2'),
HBox(children=(IntProgress(value=1, bar_style='info', description='Validating', layout=Layout(flex='2')
HBox(children=(IntProgress(value=1, bar_style='info', description='Testing', layout=Layout(flex='2'), m
TEST RESULTS
{'avg_test_acc': tensor(0.8508, dtype=torch.float64)}
```

4.5 Number of parameters

```
[0]: nparam(model, verbose=False) del model
```

5. Training

5.1 Start TensorBoard

[23]: %tensorboard --logdir logs

<IPython.core.display.Javascript object>

5.2 Training Loop

```
[24]: # network instantiation
     model = BertFinetuner(hparams=Namespace(**hyperparameters))
     #-----
     # logger configuration
     tensorboard_path = 'logs'
                                                 # set directory name
     os.makedirs(tensorboard_path, exist_ok=True) # create path
     tensorboard_logger = TensorBoardLogger(  # save logs in experiment dir
         tensorboard_path, hyperparameters['experiment_name'])
     # early stopping configuration
     early_stop = EarlyStopping(
                                              # variable to be monitored
         monitor = 'avg_val_loss',
         patience = hyperparameters['patience'], # patience
         verbose = False,
                                               # quietly
         mode = 'min'
                                               # loss should decrease
     )
     # checkpoint configuration
     ckpt_path = os.path.join('logs', hyperparameters['experiment_name'],_
      checkpoint_callback = ModelCheckpoint(
         prefix = hyperparameters['experiment_name'],  # checkpoint name prefix
         filepath = ckpt_path,
                                                     # path to checkpoint
         monitor = 'avg_val_loss',
         mode = 'min'
      #______
      # define trainer
     trainer_normal = pl.Trainer(
         max_epochs = hyperparameters['max_epochs'],
         gpus = 1,
         precision = 32,
         logger = tensorboard_logger,
         early_stop_callback = early_stop,
         checkpoint_callback = checkpoint_callback,
         resume_from_checkpoint = None,
progress_bar_refresh_rate = 50
                                                   # used to load from checkpoint (.ckpt)
                                                   # tqdm update rate (lower values can
      →cause overhead)
     )
     #-----
     # print hyperparameters
     print('Hyperparameters:\n')
     for key, value in hyperparameters.items():
         print(f'{key}: {value}')
```

```
INFO:transformers.tokenization_utils:loading file https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-vocab.txt from cache at /root/.cache/torch/transformers/26bc1ad6c0ac742e9b52263248f6d0f000 68293b33709fae12320c0e35ccfbbb.542ce4285a40d23a559526243235df47c5f75c197f04f37d1 a0c124c32c9a084
INFO:transformers.configuration_utils:loading configuration file https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
```

config.json from cache at /root/.cache/torch/transformers/4dad0251492946e18ac392
90fcfe91b89d370fee250efe9521476438fe8ca185.7156163d5fdc189c3016baca0775ffce23078
9d7fa2a42ef516483e4ca884517

```
INFO:transformers.configuration_utils:Model config BertConfig {
  "_num_labels": 2,
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "bad_words_ids": null,
  "bos_token_id": null,
  "decoder_start_token_id": null,
  "do_sample": false,
  "early_stopping": false,
  "eos_token_id": null,
  "finetuning_task": null,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "id2label": {
   "0": "LABEL_0",
   "1": "LABEL_1"
  },
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "is_decoder": false,
  "is_encoder_decoder": false,
  "label2id": {
   "LABEL_0": 0,
    "LABEL_1": 1
  },
  "layer_norm_eps": 1e-12,
  "length_penalty": 1.0,
  "max_length": 20,
  "max_position_embeddings": 512,
  "min_length": 0,
  "model_type": "bert",
  "no_repeat_ngram_size": 0,
  "num_attention_heads": 12,
  "num_beams": 1,
  "num_hidden_layers": 12,
  "num_return_sequences": 1,
  "output_attentions": false,
  "output_hidden_states": false,
  "output_past": true,
  "pad_token_id": 0,
  "prefix": null,
  "pruned_heads": {},
  "repetition_penalty": 1.0,
  "task_specific_params": null,
  "temperature": 1.0,
  "top_k": 50,
  "top_p": 1.0,
  "torchscript": false,
```

```
"use_bfloat16": false,
       "vocab_size": 30522
     }
     INFO:transformers.modeling_utils:loading weights file
     https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-
     pytorch_model.bin from cache at /root/.cache/torch/transformers/aa1ef1aede4482d0
     dbcd4d52baad8ae300e60902e88fcb0bebdec09afd232066.36ca03ab34a1a5d5fa7bc3d03d55c4f
     a650fed07220e2eeebc06ce58d0e9a157
     INFO:lightning:GPU available: True, used: True
     INFO:lightning:CUDA_VISIBLE_DEVICES: [0]
     Hyperparameters:
     experiment_name: BERT_v1
     max_epochs: 10
     patience: 1
     dataset_class: <class '__main__.IMDbDataset'>
     split_train_val: 0.8
     batch_size: 8
     nworkers: 4
     tokenizer: BERT
     num_classes: 2
     max_length: 512
     loss_func: CE
     opt_name: Adam
     lr: 1e-05
     scheduling_factor: 0.95
     manual_seed: 42
[25]: trainer_normal.fit(model)
     INFO:lightning:
                                                                                  Ι
         | Name
                                                              | Type
     Params
        | loss_func
                                                              | CrossEntropyLoss | 0
                                                              | BertModel
         | model
     109 M
     2
         | model.embeddings
                                                              | BertEmbeddings
                                                                                  | 23
     М
     3
         | model.embeddings.word_embeddings
                                                              | Embedding
                                                                                  | 23
     М
     4
         | model.embeddings.position_embeddings
                                                              | Embedding
     393 K
         | model.embeddings.token_type_embeddings
                                                              | Embedding
     5
                                                                                  | 1
     K
     6
         | model.embeddings.LayerNorm
                                                              | LayerNorm
     K
     7
         | model.embeddings.dropout
                                                              Dropout
                                                                                  1 0
     8
         | model.encoder
                                                              BertEncoder
                                                                                  | 85
     М
```

"type_vocab_size": 2,

9 M	I	model.encoder.layer	I	ModuleList	I	85
10 M	I	model.encoder.layer.0	I	BertLayer	I	7
11 M	1	model.encoder.layer.0.attention	I	BertAttention	١	2
12 M		model.encoder.layer.0.attention.self	I	BertSelfAttention	١	1
13 590		model.encoder.layer.0.attention.self.query	I	Linear	١	
14 590		model.encoder.layer.0.attention.self.key	I	Linear	I	
15 590		model.encoder.layer.0.attention.self.value	I	Linear	1	
16	-1	model.encoder.layer.0.attention.self.dropout	- 1	Dropout	1	0
17 592	1			BertSelfOutput	Ì	
18 590		model.encoder.layer.0.attention.output.dense	I	Linear	I	
19 K	1	model.encoder.layer.0.attention.output.LayerNorm	I	LayerNorm	١	1
20	-1	model.encoder.layer.0.attention.output.dropout	- 1	Dropout	1	0
21 M	1	model.encoder.layer.0.intermediate	١	BertIntermediate	١	2
22 M	1	model.encoder.layer.0.intermediate.dense	I	Linear	1	2
23 M	1	model.encoder.layer.0.output	I	BertOutput	1	2
24 M		model.encoder.layer.0.output.dense	I	Linear	1	2
25 K		model.encoder.layer.O.output.LayerNorm		LayerNorm	1	1
26		model.encoder.layer.0.output.dropout	-	Dropout		0
27 M		model.encoder.layer.1		BertLayer	I	7
28 M	1	model.encoder.layer.1.attention	-	BertAttention	1	_
29 M	1	model.encoder.layer.1.attention.self		BertSelfAttention	 -	1
30 590	K	model.encoder.layer.1.attention.self.query		Linear	 -	
31 590	K	model.encoder.layer.1.attention.self.key		Linear	 -	
32 590	K	model.encoder.layer.1.attention.self.value		Linear		
33	1	model.encoder.layer.1.attention.self.dropout		Dropout	ı	0
34 592	K	model.encoder.layer.1.attention.output	I	BertSelfOutput	١	
35 590	K	model.encoder.layer.1.attention.output.dense		Linear	١	
36 K		model.encoder.layer.1.attention.output.LayerNorm		LayerNorm	1	1
37		model.encoder.layer.1.attention.output.dropout		Dropout		0
38	I	model.encoder.layer.1.intermediate	I	BertIntermediate		2

м						
M 39 M	I	model.encoder.layer.1.intermediate.dense	I	Linear	١	2
40 M	I	model.encoder.layer.1.output	I	BertOutput	I	2
41 M	I	model.encoder.layer.1.output.dense	I	Linear	I	2
42 K	I	model.encoder.layer.1.output.LayerNorm	I	LayerNorm	I	1
43 44 M		<pre>model.encoder.layer.1.output.dropout model.encoder.layer.2</pre>		Dropout BertLayer	- 1	0 7
45 M	١	model.encoder.layer.2.attention	I	BertAttention	١	2
46 M	١	model.encoder.layer.2.attention.self	I	BertSelfAttention	١	1
47 590		model.encoder.layer.2.attention.self.query	I	Linear	١	
48 590	1	model.encoder.layer.2.attention.self.key	I	Linear	١	
49 590	 	model.encoder.layer.2.attention.self.value	I	Linear	١	
50 51	Ī	model.encoder.layer.2.attention.self.dropout model.encoder.layer.2.attention.output		Dropout BertSelfOutput	 	0
592 52	K	model.encoder.layer.2.attention.output.dense		Linear	1	
590		•			'	
53 K	1	model.encoder.layer.2.attention.output.LayerNorm	1	LayerNorm		1
54 55		model.encoder.layer.2.attention.output.dropout model.encoder.layer.2.intermediate		Dropout BertIntermediate	1	0
M	'	model.encodel.layel.2.lintermediate	'	Deltintermediate	'	2
56 M	1	model.encoder.layer.2.intermediate.dense	I	Linear	1	2
57 M	1	model.encoder.layer.2.output	1	BertOutput	1	2
58 M	1	model.encoder.layer.2.output.dense	1	Linear	1	2
59 K	1	model.encoder.layer.2.output.LayerNorm	I	LayerNorm	1	1
60		model.encoder.layer.2.output.dropout		Dropout		0
61 M	ı	model.encoder.layer.3	ı	BertLayer	ı	7
62 M	I	model.encoder.layer.3.attention	1	BertAttention	١	2
63 M	1	model.encoder.layer.3.attention.self	I	BertSelfAttention	١	1
64 590		model.encoder.layer.3.attention.self.query	I	Linear	I	
65 590	1	model.encoder.layer.3.attention.self.key	I	Linear	I	
66 590	1	model.encoder.layer.3.attention.self.value	I	Linear	١	
67	١	model.encoder.layer.3.attention.self.dropout	I	Dropout	I	0

68 592		model.encoder.layer.3.attention.output	I	BertSelfOutput	I	
69 590		model.encoder.layer.3.attention.output.dense	I	Linear	I	
70 K	1	model.encoder.layer.3.attention.output.LayerNorm	I	LayerNorm	I	1
71 72 M		<pre>model.encoder.layer.3.attention.output.dropout model.encoder.layer.3.intermediate</pre>		Dropout BertIntermediate	1	0 2
73 M	I	model.encoder.layer.3.intermediate.dense	I	Linear	I	2
74 M	1	model.encoder.layer.3.output	I	BertOutput	I	2
75 M	I	model.encoder.layer.3.output.dense	I	Linear	I	2
76 K	I	model.encoder.layer.3.output.LayerNorm	I	LayerNorm	I	1
77 78		<pre>model.encoder.layer.3.output.dropout model.encoder.layer.4</pre>		Dropout BertLayer	1	0 7
M 79	I	model.encoder.layer.4.attention	I	BertAttention	I	2
M 80 M	I	model.encoder.layer.4.attention.self	I	BertSelfAttention	I	1
81 590		model.encoder.layer.4.attention.self.query	I	Linear	I	
82 590	I	model.encoder.layer.4.attention.self.key	I	Linear	I	
83 590		model.encoder.layer.4.attention.self.value	Ι	Linear	I	
84	1	model.encoder.layer.4.attention.self.dropout	-	Dropout	1	0
85	1	model.encoder.layer.4.attention.output	-	BertSelfOutput	-	
592		•		•		
86 590		model.encoder.layer.4.attention.output.dense	I	Linear	I	
87 K		model.encoder.layer.4.attention.output.LayerNorm	I	LayerNorm	1	1
88		model.encoder.layer.4.attention.output.dropout	-	Dropout		0
89		model.encoder.layer.4.intermediate		${\tt BertIntermediate}$		2
M 90	I	model.encoder.layer.4.intermediate.dense	I	Linear	I	2
M 91 M	I	model.encoder.layer.4.output	I	BertOutput	I	2
92 M	I	model.encoder.layer.4.output.dense	I	Linear	I	2
93 K	I	model.encoder.layer.4.output.LayerNorm	I	LayerNorm	I	1
94	1	model.encoder.layer.4.output.dropout	1	Dropout	Ι	0
95 M		model.encoder.layer.5		BertLayer		7
96 M	I	model.encoder.layer.5.attention	١	BertAttention	I	2
97	I	model.encoder.layer.5.attention.self	١	${\tt BertSelfAttention}$	١	1

```
98 | model.encoder.layer.5.attention.self.query
                                                         | Linear
                                                                             1
590 K
99 | model.encoder.layer.5.attention.self.key
                                                         | Linear
                                                                             1
590 K
100 | model.encoder.layer.5.attention.self.value
                                                         | Linear
101 | model.encoder.layer.5.attention.self.dropout
                                                         | Dropout
                                                                             10
102 | model.encoder.layer.5.attention.output
                                                         | BertSelfOutput
592 K
103 | model.encoder.layer.5.attention.output.dense
                                                         | Linear
590 K
104 | model.encoder.layer.5.attention.output.LayerNorm
                                                         | LaverNorm
                                                                             1 1
105 | model.encoder.layer.5.attention.output.dropout
                                                         | Dropout
106 | model.encoder.layer.5.intermediate
                                                         | BertIntermediate
107 | model.encoder.layer.5.intermediate.dense
                                                                             1 2
                                                         | Linear
108 | model.encoder.layer.5.output
                                                         | BertOutput
                                                                             1 2
109 | model.encoder.layer.5.output.dense
                                                         | Linear
                                                                             1 2
110 | model.encoder.layer.5.output.LayerNorm
                                                         | LaverNorm
                                                                             1 1
111 | model.encoder.layer.5.output.dropout
                                                         | Dropout
112 | model.encoder.layer.6
                                                         | BertLayer
                                                                              1 7
                                                                             1 2
113 | model.encoder.layer.6.attention
                                                         | BertAttention
114 | model.encoder.layer.6.attention.self
                                                         | BertSelfAttention | 1
115 | model.encoder.layer.6.attention.self.query
                                                         | Linear
590 K
116 | model.encoder.layer.6.attention.self.key
                                                         Linear
590 K
117 | model.encoder.layer.6.attention.self.value
                                                         | Linear
590 K
118 | model.encoder.layer.6.attention.self.dropout
                                                         | Dropout
                                                                             10
119 | model.encoder.layer.6.attention.output
                                                         | BertSelfOutput
120 | model.encoder.layer.6.attention.output.dense
                                                         | Linear
121 | model.encoder.layer.6.attention.output.LayerNorm
                                                         | LayerNorm
                                                                             | 1
122 | model.encoder.layer.6.attention.output.dropout
                                                         | Dropout
                                                                             1 0
123 | model.encoder.layer.6.intermediate
                                                         | BertIntermediate
124 | model.encoder.layer.6.intermediate.dense
                                                         | Linear
                                                                             1 2
125 | model.encoder.layer.6.output
                                                         | BertOutput
                                                                             1 2
126 | model.encoder.layer.6.output.dense
                                                         | Linear
                                                                             1 2
М
```

127 K	I	model.encoder.layer.6.output.LayerNorm	I	LayerNorm	I	1
		<pre>model.encoder.layer.6.output.dropout model.encoder.layer.7</pre>		Dropout BertLayer		0 7
	I	model.encoder.layer.7.attention	١	BertAttention	I	2
	I	model.encoder.layer.7.attention.self	I	BertSelfAttention	I	1
132 590		model.encoder.layer.7.attention.self.query	I	Linear	١	
133 590		model.encoder.layer.7.attention.self.key	1	Linear	I	
134 590		model.encoder.layer.7.attention.self.value	1	Linear	1	
		model.encoder.layer.7.attention.self.dropout model.encoder.layer.7.attention.output		Dropout BertSelfOutput	1	0
	١	model.encoder.layer.7.attention.output.dense	I	Linear	I	
590 138 K		model.encoder.layer.7.attention.output.LayerNorm	1	LayerNorm	I	1
	Ī	model.encoder.layer.7.attention.output.dropout	ı	Dropout	ı	0
140 M	I	model.encoder.layer.7.intermediate	I	BertIntermediate	I	2
141 M	I	model.encoder.layer.7.intermediate.dense	1	Linear	1	2
142 M	1	model.encoder.layer.7.output	I	BertOutput	1	2
M		model.encoder.layer.7.output.dense		Linear	1	2
K		model.encoder.layer.7.output.LayerNorm		LayerNorm	1	1
		model.encoder.layer.7.output.dropout		Dropout BertLayer		0 7
140 M	1	model.encoder.layer.8	1	bertlayer	1	,
147 M	I	model.encoder.layer.8.attention	1	BertAttention	I	2
148 M	1	model.encoder.layer.8.attention.self	1	BertSelfAttention	1	1
149 590		model.encoder.layer.8.attention.self.query	I	Linear	I	
150 590		model.encoder.layer.8.attention.self.key	I	Linear	1	
590	K	model.encoder.layer.8.attention.self.value	1	Linear	1	
		model.encoder.layer.8.attention.self.dropout		Dropout		0
153 592		model.encoder.layer.8.attention.output	1	BertSelfOutput	I	
154 590		model.encoder.layer.8.attention.output.dense	I	Linear	I	
155 K		model.encoder.layer.8.attention.output.LayerNorm	1	LayerNorm	١	1
156	I	model.encoder.layer.8.attention.output.dropout	I	Dropout	١	0

157 M	model.encoder.layer.8.intermediate	1	BertIntermediate	١	2
158 M	model.encoder.layer.8.intermediate.dense	1	Linear	1	2
159 M	model.encoder.layer.8.output	I	BertOutput	1	2
160 M	model.encoder.layer.8.output.dense	1	Linear	I	2
161 K	model.encoder.layer.8.output.LayerNorm	1	LayerNorm	1	1
162	model.encoder.layer.8.output.dropout	1	Dropout	1	0
163 M	model.encoder.layer.9	I	BertLayer		7
M	model.encoder.layer.9.attention		BertAttention		2
M	model.encoder.layer.9.attention.self		BertSelfAttention		1
590 I			Linear	l	
590 I		I	Linear	١	
168 590 I	model.encoder.layer.9.attention.self.value	I	Linear	1	
169	model.encoder.layer.9.attention.self.dropout	1	Dropout		0
170	model.encoder.layer.9.attention.output	1	BertSelfOutput		
592 I	X				
590 I			Linear	1	
K	model.encoder.layer.9.attention.output.LayerNo		LayerNorm	1	1
	model.encoder.layer.9.attention.output.dropout		Dropout	-	0
M	model.encoder.layer.9.intermediate		BertIntermediate		2
M	model.encoder.layer.9.intermediate.dense		Linear		2
M	model.encoder.layer.9.output		BertOutput	·	2
M	model.encoder.layer.9.output.dense		Linear		2
K	model.encoder.layer.9.output.LayerNorm		LayerNorm		1
	model.encoder.layer.9.output.dropout		Dropout		0
	model.encoder.layer.10	I	BertLayer	ı	7
M 181	model.encoder.layer.10.attention	1	BertAttention	ı	2
M					
182 M	model.encoder.layer.10.attention.self	I	BertSelfAttention	1	1
183 590 I	model.encoder.layer.10.attention.self.query	I	Linear	١	
184 590 I	model.encoder.layer.10.attention.self.key	1	Linear	1	
185 590 I	model.encoder.layer.10.attention.self.value	I	Linear	١	

186 model.encoder.layer.10.attention.self.dropout 187 model.encoder.layer.10.attention.output 592 K		Dropout BertSelfOutput	1	0
188 model.encoder.layer.10.attention.output.dense	I	Linear	I	
189 model.encoder.layer.10.attention.output.LayerNorm	I	LayerNorm	I	1
190 model.encoder.layer.10.attention.output.dropout 191 model.encoder.layer.10.intermediate M		Dropout BertIntermediate	1	0
192 model.encoder.layer.10.intermediate.dense	I	Linear		2
193 model.encoder.layer.10.output M	I	BertOutput	1	2
194 model.encoder.layer.10.output.dense	I	Linear	1	2
195 model.encoder.layer.10.output.LayerNorm	I	LayerNorm	1	1
196 model.encoder.layer.10.output.dropout 197 model.encoder.layer.11 M		Dropout BertLayer		0 7
198 model.encoder.layer.11.attention		BertAttention	I	2
199 model.encoder.layer.11.attention.self	I	BertSelfAttention	1	1
200 model.encoder.layer.11.attention.self.query 590 K	I	Linear	1	
201 model.encoder.layer.11.attention.self.key 590 K	I	Linear	1	
202 model.encoder.layer.11.attention.self.value 590 K	I	Linear	1	
203 model.encoder.layer.11.attention.self.dropout 204 model.encoder.layer.11.attention.output 592 K		Dropout BertSelfOutput	 	0
205 model.encoder.layer.11.attention.output.dense 590 K	I	Linear	١	
206 model.encoder.layer.11.attention.output.LayerNorm	I	LayerNorm		1
207 model.encoder.layer.11.attention.output.dropout 208 model.encoder.layer.11.intermediate M		Dropout BertIntermediate		0
209 model.encoder.layer.11.intermediate.dense	I	Linear		2
210 model.encoder.layer.11.output	١	BertOutput	I	2
211 model.encoder.layer.11.output.dense	I	Linear	1	2
212 model.encoder.layer.11.output.LayerNorm	I	LayerNorm	1	1
213 model.encoder.layer.11.output.dropout 214 model.pooler 590 K		Dropout BertPooler	 	0
215 model.pooler.dense 590 K	I	Linear	I	

[25]: 1

6. Evaluation

```
[26]: trainer_normal.test(model)

HBox(children=(IntProgress(value=1, bar_style='info', description='Testing', layout=Layout(flex='2'), moderate to the state of the state of
```

End of the notebook