

November 27, 2022

## 1 TASK 2

## 2 NN with pre\_trained Embeddings

**Objective:** - Learn to use pre-trained embeddings in a Neural Network

## 3 Specify Project Folders

```
[1]: # This is the path where we will download and save data
from pathlib import Path
if 'google.colab' in str(get_ipython()):
    base_folder = Path('/content/drive/MyDrive/NLP')
else:
    base_folder = Path('/content/drive/MyDrive/NLP')
```

```
[2]: custom_functions = base_folder/'data/customfiles'
```

```
[3]: import sys
sys.path.append(str(custom_functions))
```

```
[4]: sys.path
```

```
[4]: ['/content',
      '/env/python',
      '/usr/lib/python3.7.zip',
      '/usr/lib/python3.7',
      '/usr/lib/python3.7/lib-dynload',
      '',
      '/usr/local/lib/python3.7/dist-packages',
      '/usr/lib/python3/dist-packages',
      '/usr/local/lib/python3.7/dist-packages/IPython/extensions',
      '/root/.ipython',
      '/content/drive/MyDrive/NLP/data/customfiles']
```

## 4 Import libraries

```
[5]: %load_ext autoreload
      %autoreload 2
```

```
[6]: if 'google.colab' in str(get_ipython()):
      print('Running on CoLab')
      else:
      print('Not running on CoLab')
```

Running on CoLab

```
[7]: if 'google.colab' in str(get_ipython()):
      !pip install --upgrade gensim -qq
```

```
|                | 24.1 MB 19.7 MB/s
```

```
[8]: if 'google.colab' in str(get_ipython()):
      !pip install wandb --upgrade
```

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>

Collecting wandb

Downloading wandb-0.13.5-py2.py3-none-any.whl (1.9 MB)

```
|                | 1.9 MB 13.9 MB/s
```

Collecting GitPython>=1.0.0

Downloading GitPython-3.1.29-py3-none-any.whl (182 kB)

```
|                | 182 kB 70.4 MB/s
```

Requirement already satisfied: psutil>=5.0.0 in

/usr/local/lib/python3.7/dist-packages (from wandb) (5.4.8)

Requirement already satisfied: Click!=8.0.0,>=7.0 in

/usr/local/lib/python3.7/dist-packages (from wandb) (7.1.2)

Collecting docker-pycreds>=0.4.0

Downloading docker\_pycreds-0.4.0-py2.py3-none-any.whl (9.0 kB)

Requirement already satisfied: protobuf!=4.0.\*,!=4.21.0,<5,>=3.12.0 in

/usr/local/lib/python3.7/dist-packages (from wandb) (3.19.6)

Collecting setproctitle

Downloading setproctitle-1.3.2-cp37-cp37m-manylinux\_2\_5\_x86\_64.manylinux1\_x86\_64.manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (30 kB)

Collecting sentry-sdk>=1.0.0

Downloading sentry\_sdk-1.11.1-py2.py3-none-any.whl (168 kB)

```
|                | 168 kB 51.4 MB/s
```

Requirement already satisfied: requests<3,>=2.0.0 in

/usr/local/lib/python3.7/dist-packages (from wandb) (2.23.0)

Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from wandb) (57.4.0)

Requirement already satisfied: promise<3,>=2.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (2.3)

Collecting pathtools

  Downloading pathtools-0.1.2.tar.gz (11 kB)

Collecting shortuuid>=0.5.0

  Downloading shortuuid-1.0.11-py3-none-any.whl (10 kB)

Requirement already satisfied: six>=1.13.0 in /usr/local/lib/python3.7/dist-packages (from wandb) (1.15.0)

Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from wandb) (6.0)

Collecting gitdb<5,>=4.0.1

  Downloading gitdb-4.0.10-py3-none-any.whl (62 kB)

    |                          | 62 kB 1.1 MB/s

Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.7/dist-packages (from GitPython>=1.0.0->wandb) (4.1.1)

Collecting smmap<6,>=3.0.1

  Downloading smmap-5.0.0-py3-none-any.whl (24 kB)

Requirement already satisfied: urllib3!=1.25.0,!<1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.0.0->wandb) (1.24.3)

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.0.0->wandb) (3.0.4)

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.0.0->wandb) (2.10)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.0.0->wandb) (2022.9.24)

Collecting sentry-sdk>=1.0.0

  Downloading sentry\_sdk-1.11.0-py2.py3-none-any.whl (168 kB)

    |                          | 168 kB 58.3 MB/s

  Downloading sentry\_sdk-1.10.1-py2.py3-none-any.whl (166 kB)

    |                          | 166 kB 54.0 MB/s

  Downloading sentry\_sdk-1.10.0-py2.py3-none-any.whl (166 kB)

    |                          | 166 kB 50.2 MB/s

  Downloading sentry\_sdk-1.9.10-py2.py3-none-any.whl (162 kB)

    |                          | 162 kB 50.4 MB/s

  Downloading sentry\_sdk-1.9.9-py2.py3-none-any.whl (162 kB)

    |                          | 162 kB 27.6 MB/s

  Downloading sentry\_sdk-1.9.8-py2.py3-none-any.whl (158 kB)

    |                          | 158 kB 54.0 MB/s

  Downloading sentry\_sdk-1.9.7-py2.py3-none-any.whl (157 kB)

    |                          | 157 kB 54.0 MB/s

  Downloading sentry\_sdk-1.9.6-py2.py3-none-any.whl (157 kB)

    |                          | 157 kB 14.3 MB/s

  Downloading sentry\_sdk-1.9.5-py2.py3-none-any.whl (157 kB)

    |                          | 157 kB 59.3 MB/s

  Downloading sentry\_sdk-1.9.4-py2.py3-none-any.whl (157 kB)

    |                          | 157 kB 61.7 MB/s

  Downloading sentry\_sdk-1.9.3-py2.py3-none-any.whl (157 kB)

```

| 157 kB 53.2 MB/s
Downloading sentry_sdk-1.9.2-py2.py3-none-any.whl (157 kB)
| 157 kB 68.3 MB/s
Downloading sentry_sdk-1.9.1-py2.py3-none-any.whl (157 kB)
| 157 kB 10.4 MB/s
Downloading sentry_sdk-1.9.0-py2.py3-none-any.whl (156 kB)
| 156 kB 55.8 MB/s
Building wheels for collected packages: pathtools
Building wheel for pathtools (setup.py) ... done
Created wheel for pathtools: filename=pathtools-0.1.2-py3-none-any.whl
size=8806
sha256=df94becf4248861d0589c52b613cc855c20ef47824c7a181c4975ed54ff8c742
Stored in directory: /root/.cache/pip/wheels/3e/31/09/fa59cef12cdcfec627b3d24
273699f390e71828921b2cbba2
Successfully built pathtools
Installing collected packages: smmap, gitdb, shortuuid, setproctitle, sentry-
sdk, pathtools, GitPython, docker-pycreds, wandb
Successfully installed GitPython-3.1.29 docker-pycreds-0.4.0 gitdb-4.0.10
pathtools-0.1.2 sentry-sdk-1.9.0 setproctitle-1.3.2 shortuuid-1.0.11 smmap-5.0.0
wandb-0.13.5

```

```

[9]: if 'google.colab' in str(get_ipython()):
    from google.colab import drive
    drive.mount('/content/drive')

```

Mounted at /content/drive

```

[10]: # Import random function

import torch
import torch.nn as nn
import torch.nn.functional as F
from torchtext.vocab import vocab
from torch.optim.lr_scheduler import ReduceLROnPlateau, OneCycleLR, StepLR

import wandb
import spacy
#import custom_preprocessor as cp

import random
from datetime import datetime
import numpy as np
from pathlib import Path
import pandas as pd
import joblib
from collections import Counter

```

```

from pathlib import Path

from sklearn.model_selection import train_test_split

import gensim
from gensim.models import KeyedVectors

from types import SimpleNamespace

```

```
[11]: gensim.__version__
```

```
[11]: '4.2.0'
```

```

[12]: data_folder = base_folder/'          '/datasets'
      save_model_folder = base_folder/'          '/models'

```

We will be using W&B for visualization.

```

[13]: # Login to W&B
      wandb.login()

```

ERROR:wandb.jupyter:Failed to detect the name of this notebook, you can set it manually with the WANDB\_NOTEBOOK\_NAME environment variable to enable code saving.

<IPython.core.display.Javascript object>

wandb: Appending key for api.wandb.ai to your netrc file:  
/root/.netrc

```
[13]: True
```

## 4.1 Train/Test/Valid Dataset

```
[14]: df = joblib.load(data_folder/'df_multilabel_ _cleaned.joblib')
```

```
[15]: df.head()
```

```

[15]:   Unnamed: 0  Unnamed: 0.1      Id \
0         35264         3589944  3589945
1         56592         5756414  5756415
2         23303         2358596  2358597
3         42563         4332881  4332882
4         58216         5922132  5922133

```

Title \

```

0          ASP Query String From DropDown
1  How can I run JavaScript code at server side J...
2  linq to sql throwing an exception row not foun...
3          Running a Python script on a PHP server
4  some advice on how to write a window.resize fu...

```

	Body	Tags \
0	<p>I have a webpage: <strong>Menu.aspx</strong>...	c# asp.net
1	<p>I want to run JavaScript code at the server...	java javascript
2	<p>Hi I am linq to sql and i am getting the er...	c# asp.net
3	<p>I am running a nginx web server, along with...	php python
4	<p>Im trying to write a function that resizes ...	javascript jquery

	Tag_Number	combined_text \
0	[0, 9]	ASP Query String From DropDown <p>I have a web...
1	[1, 3]	How can I run JavaScript code at server side J...
2	[0, 9]	linq to sql throwing an exception row not foun...
3	[2, 7]	Running a Python script on a PHP server <p>I a...
4	[3, 5]	some advice on how to write a window.resize fu...

```

          cleaned_text_lemma_stop_removal
0  asp query string dropdown webpage following co...
1  run javascript code server java code want run ...
2  linq sql throw exception row find change hi li...
3  run python script php server run nginx web ser...
4  advice write function m try write function res...

```

```
[16]: X = df['cleaned_text_lemma_stop_removal'].values
```

```
[17]: y = df['Tag_Number'].values
```

```
[18]: # import swifter
      from ast import literal_eval
      df['Tag_Number_list'] = df['Tag_Number'].apply(lambda x: literal_eval(x))
```

```
[19]: y_final = df['Tag_Number_list'].values
```

```
[20]: from sklearn.preprocessing import MultiLabelBinarizer as mlb
```

```
[21]: y_one_hot = mlb().fit_transform(y_final)
```

```
[22]: y_one_hot[0]
```

```
[22]: array([1, 0, 0, 0, 0, 0, 0, 0, 0, 1])
```

```
[23]: y_one_hot.shape
```

```
[23]: (47427, 10)
```

```
[24]: X_train, X_valid_test, y_train, y_valid_test = train_test_split(X, y_one_hot,
    ↪test_size = 0.40, random_state=42)
```

```
[25]: X_valid, X_test, y_valid, y_test = train_test_split(X_valid_test, y_valid_test,
    ↪test_size = 0.50, random_state=42)
```

```
[26]: print(len(X_train), len(X_valid), len(X_test))
```

```
28456 9485 9486
```

```
[27]: X_test.shape
```

```
[27]: (9486,)
```

## 4.2 Data PreProcessing

We will use the preprocessed files we created earlier.

```
[28]: print(type(X_train))
      print(type(y_train))
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

## 4.3 Custom Dataset Class

```
[29]: class CustomDataset(torch.utils.data.Dataset):
      """MultiLabel dataset."""

      def __init__(self, X, y):
          self.X = np.array(X)
          self.y = y

      def __len__(self):
          return len(self.X)

      def __getitem__(self, idx):
          if torch.is_tensor(idx):
              idx = idx.tolist()

          text = self.X[idx]
          labels = self.y[idx]
          sample = (text, labels)
```

```
return sample
```

```
[30]: trainset = CustomDataset(X_train,y_train)
      validset = CustomDataset(X_valid,y_valid)
      testset = CustomDataset(X_test,y_test)
```

```
[31]: trainset.__getitem__([11])
```

```
[31]: (array(['code multiple button navigation java activity question 1 2 activity
wonder optimize create 2 activity multiple listener create multiple java file
button(onclick listener question 2 try create multiple listener java button work
syntax multiple listener java file update code issue matter button click lead
page      package import activity import context import intent import bundle
import button import view import view onclicklistener  public class activity1
extend activity2  button button1 button button2 button button3 button button4
button button5 button button6  public void onCreate(bundle savedInstanceState
super.onCreate(savedInstanceState      setContentView(r.layout.fineline
addlisteneronbutton      public void addlisteneronbutton  final context context
=  button1 = button findViewById(r.id.autobody  button1.setonclicklistener(new
onclicklistener      public void onclick(view arg0      intent intent =
new intent(context      startactivity(intent      button2 = button
findViewById(r.id.glass  button2.setonclicklistener(new onclicklistener
override      public void onclick(view arg0      intent intent = new
intent(context      startactivity(intent      button3 = button
findViewById(r.id.wheels  button3.setonclicklistener(new onclicklistener
override      public void onclick(view arg0      intent intent = new
intent(context      startactivity(intent      button4 = button
findViewById(r.id.speedy  button4.setonclicklistener(new onclicklistener
override      public void onclick(view arg0      intent intent = new
intent(context      startactivity(intent      button5 = button
findViewById(r.id.sevan  button5.setonclicklistener(new onclicklistener
override      public void onclick(view arg0      intent intent = new
intent(context      startactivity(intent      button6 = button
findViewById(r.id.towing button6.setonclicklistener(new onclicklistener override
public void onclick(view arg0      intent intent = new intent(context
startactivity(intent      package  import activity import bundle import
button  public class activity2 extend activity  button button1  public void
oncreate1(bundle savedInstanceState super.onCreate(savedInstanceState
setcontentview(r.layout.autobody button button2  public void onCreate2(bundle
savedInstanceState super.onCreate(savedInstanceState
setcontentview(r.layout.glass button button3  public void onCreate3(bundle
savedInstanceState super.onCreate(savedInstanceState
setcontentview(r.layout.wheel button button4  public void onCreate(bundle
savedInstanceState super.onCreate(savedInstanceState
setcontentview(r.layout.speedy button button5  public void onCreate5(bundle
savedInstanceState super.onCreate(savedInstanceState
setcontentview(r.layout.sevan  button button6  public void onCreate6(bundle
```



```

savedinstancestate super.oncreate(savedinstancestate
setcontentview(r.layout.towe    '],
dtype=object), array([[0, 1, 0, 0, 1, 0, 0, 0, 0, 0]]))

```

## 4.4 Create Vocab

```

[32]: def create_vocab(dataset, min_freq):
        counter = Counter()
        for (text, _) in dataset:
            counter.update(str(text).split())
        my_vocab = vocab(counter, min_freq=min_freq)
        my_vocab.insert_token('<unk>', 0)
        my_vocab.set_default_index(0)
        return my_vocab

```

vocab should always be created based on trainset

```

[33]: multilabel_vocab = create_vocab(trainset, min_freq = 2)

```

```

[34]: len(multilabel_vocab)

```

```

[34]: 90235

```

```

[35]: multilabel_vocab.get_itos()[0:5]

```

```

[35]: ['<unk>', 'rearrange', 'order', 'list', 'web']

```

```

[36]: multilabel_vocab['teffy']

```

```

[36]: 0

```

## 4.5 Collate\_fn for Data Loaders

```

[116]: # Creating a lambda function objects that will be used to get the indices of
        ↪ words from vocab
        text_pipeline = lambda x : [multilabel_vocab[token] for token in str(x).split()]
        vector = np.vectorize(np.int_)
        label_pipeline = lambda y : vector(y)

```

```

[117]: len(y)

```

```

[117]: 47427

```

```

[118]: '''
        We know that input to the embedding layers are indices of words from the vocab.

```

*The `collate_batch()` accepts batch of data and gets the indices of text from `vocab` and returns the same*

*We will include this `collate_batch()` in `collat_fn` attribute of `DataLoader`. So it will create a batch of data containing indices of words and corresponding labels.*

*But for `EmbeddingBag` we need one more extra parameter, that is `offset`. `offsets` determines the starting index position of each bag (sequence) in input.*

```
'''
def collate_batch(batch):
    label_list, text_list, offsets = [], [], [0]
    for (_text, _label) in batch:
        label_list.append(label_pipeline(_label))
        processed_text = torch.tensor(text_pipeline(_text), dtype=torch.int64)
        text_list.append(processed_text)
        offsets.append(processed_text.size(0))
    label_list = torch.tensor(label_list, dtype=torch.int64)
    offsets = torch.tensor(offsets[:-1]).cumsum(dim=0)
    text_list = torch.cat(text_list)
    return text_list, label_list, offsets
```

## 4.6 Check Data Loader

```
[119]: batch_size=2
check_loader= torch.utils.data.DataLoader(dataset=trainset,
                                           batch_size=batch_size,
                                           shuffle=True,
                                           collate_fn=collate_batch,
                                           num_workers=3)
```

```
/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py:566:
UserWarning: This DataLoader will create 3 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than
what this DataLoader is going to create. Please be aware that excessive worker
creation might get DataLoader running slow or even freeze, lower the worker
number to avoid potential slowness/freeze if necessary.
  cpuset_checked))
```

```
[120]: for text, label, offsets in check_loader:
        print(label, text, offsets)
        break
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/autograd/tensor_new.cpp:201.)
```

```

app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
tensor([[0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 1, 0, 0, 0, 1]]) tensor([16908,  425,  376,  521,
36, 1137, 1186,  460,  521,  786,
        16909, 16908, 16910,  322,  17,  688,  786, 16909,  689,  96,
        7543, 16911,  19,  17,  688,  786, 16909,  689, 16912,  19,
         68,  128, 1261, 3757,  368,  105,  163,  4, 2650, 2669,
        1125,  128,  688,  427,  55, 2190,  431,  24,  68, 5871,
        368]) tensor([ 0, 30])

```

```
[42]: # Smaller Sample
```

```

[43]: # Fix seed value
SEED = 2345
random.seed(SEED)

# We will be using 500 observations for text and 100 for valid dataset
# We do not need valid for overfitting, it will help to check errors in code
train_sample_size = 500
valid_sample_size = 100

# Getting n random indices
train_subset_indices = random.sample(range(0, len(trainset)), train_sample_size)
valid_subset_indices = random.sample(range(0, len(validset)), valid_sample_size)

# Getting subset of dataset
train_subset = torch.utils.data.Subset(trainset, train_subset_indices)
valid_subset = torch.utils.data.Subset(validset, valid_subset_indices)

```

## 5 Model for ANN

```
[44]: class MLPCustom(nn.Module):
    def __init__(self, embed_dim, vocab_size, h_sizes_list, d_prob_list,
        ↪output_dim, non_linearity, batch_norm, task, pretrained_weights):

        super().__init__()

        self.output_dim = output_dim
        self.vocab_size = vocab_size
        self.embed_dim = embed_dim
        self.h_sizes_list = h_sizes_list
        self.d_prob_list = d_prob_list
        self.batch_norm = batch_norm
        self.task = task
        self.pretrained_weights= pretrained_weights

        self.non_linearity = non_linearity

        model_layers = []

        # embedding_layer
        if self.task == 2:
            self.embedding = nn.EmbeddingBag(self.vocab_size, self.embed_dim)

        # Task 5
        if self.task == 5:
            self.embedding = nn.EmbeddingBag(vocab_size, self.embed_dim).
            ↪from_pretrained(pretrained_weights,

        ↪freeze = True)

        # Task 6
        if self.task == 6:
            self.embedding = nn.EmbeddingBag(vocab_size, self.embed_dim).
            ↪from_pretrained(pretrained_weights,

        ↪freeze = False)

        input_dim = self.embed_dim
        # hidden layers, dropout, non_linearity, batchnorm layers
        for k, hidden_size in enumerate(self.h_sizes_list):
            # hidden_layer
            model_layers.append(nn.Linear(input_dim, hidden_size))
            # Activation function
            model_layers.append(self.non_linearity)
```

```

    # Dropout Layer
    model_layers.append(nn.Dropout(p=self.d_prob_list[k]))
    # Batch_Norm Layer
    if self.batch_norm:
        model_layers.append(nn.BatchNorm1d(hidden_size, momentum = 0.9))
        input_dim = hidden_size

    # output layer
    if len(self.h_sizes_list)>0:
        model_layers.append(nn.Linear(self.h_sizes_list[-1], self.output_dim))
    else:
        model_layers.append(nn.Linear(self.embed_dim, self.output_dim))

    self.module_list = nn.ModuleList(model_layers)

def forward(self, x, offsets):
    out = self.embedding(x, offsets) # batchsize, embedding_dim
    for layer in self.module_list:
        out = layer(out)
    return out

# Note : We do not need to apply softmax as we will be using nn.
→CrossEntropy Loss

```

## 5.1 Function for Training Loops

Model Training involves five steps:

- Step 0: Randomly initialize parameters / weights
- Step 1: Compute model's predictions - forward pass
- Step 2: Compute loss
- Step 3: Compute the gradients
- Step 4: Update the parameters
- Step 5: Repeat steps 1 - 4

Model training is repeating this process over and over, for many **epochs**.

We will specify number of **epochs** and during each epoch we will iterate over the complete dataset and will keep on updating the parameters.

**Learning rate** and **epochs** are known as hyperparameters. We have to adjust the values of these two based on validation dataset.

We will now create functions for step 1 to 4.

```

[45]: def train(train_loader, loss_function, model, optimizer, grad_clipping,
    →max_norm, log_batch, log_interval):

```

```

# Training Loop

# initialize variables as global
# these counts will be updated every epoch
global batch_ct_train

# Initialize train_loss at the he start of the epoch
running_train_loss = 0
running_train_correct = 0

# put the model in training mode

model.train()
# Iterate on batches from the dataset using train_loader
for input_, targets, offsets in train_loader:

    # move inputs and outputs to GPUs
    input_ = input_.to(device)
    targets = targets.to(device)
    offsets = offsets.to(device)

    # Step 1: Forward Pass: Compute model's predictions
    output = model(input_,offsets)

    # Step 2: Compute loss
    loss = loss_function(output, targets.float())

    # Correct prediction
    # y_pred = torch.argmax(output, dim = 1)
    # correct = torch.sum(y_pred == targets)

    batch_ct_train += 1

    # Step 3: Backward pass -Compute the gradients
    optimizer.zero_grad()
    loss.backward()

    # Gradient Clipping
    if grad_clipping:
        nn.utils.clip_grad_norm_(model.parameters(), max_norm=max_norm,
↪norm_type=2)

    # Step 4: Update the parameters
    optimizer.step()

    # Add train loss of a batch

```

```

running_train_loss += loss.item()

# Add Correct counts of a batch
# running_train_correct += correct

# log batch loss and accuracy
if log_batch:
    if ((batch_ct_train + 1) % log_interval) == 0:
        wandb.log({"Train Batch Loss ": loss})
        # wandb.log({"Train Batch Acc ": correct/len(targets)})

# Calculate mean train loss for the whole dataset for a particular epoch
train_loss = running_train_loss/len(train_loader)

# Calculate accuracy for the whole dataset for a particular epoch
# train_acc = running_train_correct/len(train_loader.dataset)

return train_loss

```

## 5.2 Function for Validation Loops

[46]: `def validate(valid_loader, loss_function, model, log_batch, log_interval):`

```

# inititalize variables as global
# these counts will be updated every epoch
global batch_ct_valid

# Validation/Test loop
# Initialize valid_loss at the he strat of the epoch
running_val_loss = 0
# running_val_correct = 0

# put the model in evaluation mode
model.eval()

with torch.no_grad():
    for input_, targets, offsets in valid_loader:

        # move inputs and outputs to GPUs
        input_ = input_.to(device)
        targets = targets.to(device)
        offsets = offsets.to(device)

        # Step 1: Forward Pass: Compute model's predictions

```

```

output = model(input_,offsets)

# Step 2: Compute loss
loss = loss_function(output, targets.float())

# Correct Predictions
# y_pred = torch.argmax(output, dim = 1)
# correct = torch.sum(y_pred == targets)

batch_ct_valid += 1

# Add val loss of a batch
running_val_loss += loss.item()

# Add correct count for each batch
# running_val_correct += correct

# log batch loss and accuracy
if log_batch:
    if ((batch_ct_valid + 1) % log_interval) == 0:
        wandb.log({f"Valid Batch Loss   ": loss})
        # wandb.log({f"Valid Batch Accuracy ": correct/len(targets)})

# Calculate mean val loss for the whole dataset for a particular epoch
val_loss = running_val_loss/len(valid_loader)

# Calculate accuracy for the whole dataset for a particular epoch
# val_acc = running_val_correct/len(valid_loader.dataset)

# scheduler step
scheduler.step(val_loss)
# scheduler.step()

return val_loss

```

### 5.3 Function for Model Training

We will now create a function for step 5 of model training

```

[47]: def train_loop(train_loader, valid_loader, model, optimizer, loss_function,
    ↪ epochs, device, patience, early_stopping,
        file_model, save_best_model):

    """
    Function for training the model and plotting the graph for train & validation,
    ↪ loss vs epoch.
    """

```



*Input: iterator for train dataset, initial weights and bias, epochs, learning\_*  
*→rate, batch size.*

*Output: final weights, bias and train loss and validation loss for each epoch.*  
*"""*

*# Create lists to store train and val loss at each epoch*

train\_loss\_history = []

valid\_loss\_history = []

*# train\_acc\_history = []*

*# valid\_acc\_history = []*

*# initialize variables for early stopping*

delta = 0

best\_score = **None**

valid\_loss\_min = np.Inf

counter\_early\_stop=0

early\_stop=**False**

*# Iterate for the given number of epochs*

*# Step 5: Repeat steps 1 - 4*

**for** epoch **in** range(epochs):

    t0 = datetime.now()

*# Get train loss for one epoch*

        train\_loss = train(train\_loader, loss\_function, model, optimizer,  
                            wandb.config.GRAD\_CLIPPING, wandb.config.

→MAX\_NORM,

                            wandb.config.LOG\_BATCH, wandb.config.

→LOG\_INTERVAL)

        valid\_loss = validate(valid\_loader, loss\_function, model,  
                                wandb.config.LOG\_BATCH, wandb.config.

→LOG\_INTERVAL)

    dt = datetime.now() - t0

*# Save history of the Losses and accuracy*

        train\_loss\_history.append(train\_loss)

*# train\_acc\_history.append(train\_acc)*

        valid\_loss\_history.append(valid\_loss)

*# valid\_acc\_history.append(valid\_acc)*

*# Log the train and valid loss to wandb*

        wandb.log({"Train Loss ": train\_loss, "epoch": epoch})

```

# wandb.log({f"Train Acc ": train_acc, "epoch": epoch})

wandb.log({f"Valid Loss ": valid_loss, "epoch": epoch})
# wandb.log({f"Valid Acc ": valid_acc, "epoch": epoch})

if early_stopping:
    score = -valid_loss
    if best_score is None:
        best_score=score
        print(f'Validation loss has decreased ({valid_loss_min:.6f} -->_
↪{valid_loss:.6f}). Saving Model...')
        torch.save(model.state_dict(), file_model)
        valid_loss_min = valid_loss

    elif score < best_score + delta:
        counter_early_stop += 1
        print(f'Early stopping counter: {counter_early_stop} out of {patience}')
        if counter_early_stop > patience:
            early_stop = True

    else:
        best_score = score
        print(f'Validation loss has decreased ({valid_loss_min:.6f} -->_
↪{valid_loss:.6f}). Saving model...')
        torch.save(model.state_dict(), file_model)
        counter_early_stop=0
        valid_loss_min = valid_loss

if early_stop:
    print('Early Stopping')
    break

elif save_best_model:

    score = -valid_loss
    if best_score is None:
        best_score=score
        print(f'Validation loss has decreased ({valid_loss_min:.6f} -->_
↪{valid_loss:.6f}). Saving Model...')
        torch.save(model.state_dict(), file_model)
        valid_loss_min = valid_loss

    elif score < best_score + delta:
        print(f'Validation loss has not decreased ({valid_loss_min:.6f} -->_
↪{valid_loss:.6f}). Not Saving Model...')

```

```

    else:
        best_score = score
        print(f'Validation loss has decreased ({valid_loss_min:.6f} -->
→{valid_loss:.6f}). Saving model...')
        torch.save(model.state_dict(), file_model)
        valid_loss_min = valid_loss

    else:
        torch.save(model.state_dict(), file_model)

    # Print the train loss and accuracy for given number of epochs, batch size
→and number of samples
    print(f'Epoch : {epoch+1} / {epochs}')
    print(f'Time to complete {epoch+1} is {dt}')
    print(f'Learning rate: {scheduler._last_lr[0]}')
    print(f'Train Loss: {train_loss : .4f} ')
    print(f'Valid Loss: {valid_loss : .4f} ')
    print()
    torch.cuda.empty_cache()

return train_loss_history, valid_loss_history

```

## 5.4 Function for Accuracy and Predictions

Now we have final values for weights and bias after training the model. We will use these values to make predictions on the test dataset.

```

[48]: def get_pred(data_loader, model):
    """
    Function to get predictions for a given test set and calculate accuracy.
    Input: Iterator to the test set.
    Output: Predictions and Accuracy for test set.
    """
    model.eval()
    with torch.no_grad():
        # Array to store predicted labels
        predictions = torch.Tensor()
        predictions = predictions.to(device)

        outputs = torch.Tensor()
        outputs = outputs.to(device)

        # Array to store actual labels
        y = torch.Tensor()
        y = y.to(device)
        # Iterate over batches from test set

```

```

for input_, targets, offsets in data_loader:

    # move inputs and outputs to GPUs
    input_ = input_.to(device)
    targets = targets.to(device)
    offsets = offsets.to(device)

    # Calculated the predicted labels
    output = model(input_,offsets)
    predicted_y = output.clone()

    # Update teh output
    predicted_y[predicted_y>0] = 1
    predicted_y[predicted_y<=0] =0

    # Add the predicted labels to the array
    predictions = torch.cat((predictions, predicted_y))

    outputs = torch.cat((outputs, output))

    # Add the actual labels to the array
    y = torch.cat((y, targets))

# Return array containing predictions and accuracy
return y, predictions

```

## 6 Meta Data

```

[54]: hyperparameters = SimpleNamespace(
    EMBED_DIM = 5000,
    VOCAB_SIZE = len(multilabel_vocab),
    OUTPUT_DIM = 10,
    HIDDEN_SIZES_LIST = [500,200], # 100 layers of size 200 [200]*100
    DPROB_LIST = [0,0],
    NON_LINEARITY= nn.SELU(),
    PRETRAINED_WEIGHTS_TENSOR = 0, #torch.tensor(pretrained_weights).float(),
    BATCH_NORM = True,
    EPOCHS = 10,
    TASK = 2,
    BATCH_SIZE = 256,
    LEARNING_RATE = 0.02,
    DATASET="MultiLabel",
    ARCHITECTUREe="2_hidden_layers",
    LOG_INTERVAL = 25,
    LOG_BATCH = True,

```

```

FILE_MODEL = save_model_folder/'1_ _part_B_full.pt',
GRAD_CLIPPING = False,
MAX_NORM = 0,
MOMENTUM = 0,
PATIENCE = 10,
EARLY_STOPPING = True,
SCHEDULER_FACTOR = 0.5,
SCHEDULER_PATIENCE = 0,
WEIGHT_DECAY = 0.0005,
SAVE_BEST_MODEL = True,
DEVICE = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
)

```

```
[55]: torch.cuda.is_available()
```

```
[55]: True
```

## 7 Data Loaders, Loss Function, Optimizer

```

[56]: # Initialize a new project
import random
wandb.init(name = 'task2', project = 'NLP_ _part2', config = hyperparameters)

```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value='0.001 MB of 0.001 MB uploaded (0.000 MB deduped)\r'), FloatProgress
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

[56]: <wandb.sdk.wandb\_run.Run at 0x7fd39ec048d0>

```
[57]: wandb.config = hyperparameters
      wandb.config
```

```
[57]: namespace(ARCHITECTURE='2_hidden_layers', BATCH_NORM=True, BATCH_SIZE=256,
      DATASET='MultiLabel', DEVICE=device(type='cuda', index=0), DPROB_LIST=[0, 0],
      EARLY_STOPPING=True, EMBED_DIM=5000, EPOCHS=10, FILE_MODEL=PosixPath('/content/drive/MyDrive/NLP/-----/models/-----.pt'), GRAD_CLIPPING=False,
      HIDDEN_SIZES_LIST=[500, 200], LEARNING_RATE=0.02, LOG_BATCH=True,
      LOG_INTERVAL=25, MAX_NORM=0, MOMENTUM=0, NON_LINEARITY=SELU(), OUTPUT_DIM=10,
      PATIENCE=10, PRETRAINED_WEIGHTS_TENSOR=0, SAVE_BEST_MODEL=True,
      SCHEDULER_FACTOR=0.5, SCHEDULER_PATIENCE=0, TASK=2, VOCAB_SIZE=90235,
      WEIGHT_DECAY=0.0005)
```

```
[58]: # Fix seed value
      SEED = 2345
      random.seed(SEED)
      np.random.seed(SEED)
      torch.manual_seed(SEED)
      torch.cuda.manual_seed(SEED)
      torch.backends.cudnn.deterministic = True

      # Data Loader
      train_loader = torch.utils.data.DataLoader(trainset, batch_size=wandb.config.
      ↪BATCH_SIZE, shuffle = True,
      ↪num_workers = 4)
      valid_loader = torch.utils.data.DataLoader(validset, batch_size=wandb.config.
      ↪BATCH_SIZE, shuffle = False,
      ↪num_workers = 4)
      test_loader = torch.utils.data.DataLoader(testset, batch_size=wandb.config.
      ↪BATCH_SIZE, shuffle = False,
      ↪num_workers = 4)

      # cross entropy loss function
      loss_function = nn.BCEWithLogitsLoss()

      # model
      model_multilabel = MLPCustom(wandb.config.EMBED_DIM,
      ↪wandb.config.VOCAB_SIZE,
      ↪wandb.config.HIDDEN_SIZES_LIST,
      ↪wandb.config.DPROB_LIST,
      ↪wandb.config.OUTPUT_DIM,
      ↪wandb.config.NON_LINEARITY,
```

```

        wandb.config.BATCH_NORM,
        wandb.config.TASK,
        wandb.config.PRETRAINED_WEIGHTS_TENSOR)

model_multilabel.to(wandb.config.DEVICE)

def init_weights(m):
    if type(m) == nn.Linear:
        torch.nn.init.kaiming_normal_(m.weight)
        torch.nn.init.zeros_(m.bias)

# apply initialization recursively to all modules
# model_multilabel.apply(init_weights)

# Intialize stochastic gradient descent optimizer
optimizer = torch.optim.Adam(model_multilabel.parameters(),
                              lr = wandb.config.LEARNING_RATE,
                              weight_decay=wandb.config.WEIGHT_DECAY)

# wandb.config.OPTIMIZER = optimizer

scheduler = ReduceLROnPlateau(optimizer, mode='min', factor= wandb.config.
    ↪ SCHEDULER_FACTOR,
                              patience=wandb.config.SCHEDULER_PATIENCE,
    ↪ verbose=True)

#scheduler = StepLR(optimizer, gamma=0.4, step_size=1, verbose=True)

```

```
[59]: wandb.config.DEVICE
```

```
[59]: device(type='cuda', index=0)
```

```
[60]: wandb.config
```

```
[60]: namespace(ARCHITECTURE='2_hidden_layers', BATCH_NORM=True, BATCH_SIZE=256,
DATASET='MultiLabel', DEVICE=device(type='cuda', index=0), DPROB_LIST=[0, 0],
EARLY_STOPPING=True, EMBED_DIM=5000, EPOCHS=10, FILE_MODEL=PosixPath('/content/drive/MyDrive/NLP/FinalModel/epoch_100_part_B_full.pt'), GRAD_CLIPPING=False,
HIDDEN_SIZES_LIST=[500, 200], LEARNING_RATE=0.02, LOG_BATCH=True,
LOG_INTERVAL=25, MAX_NORM=0, MOMENTUM=0, NON_LINEARITY=SELU(), OUTPUT_DIM=10,
PATIENCE=10, PRETRAINED_WEIGHTS_TENSOR=0, SAVE_BEST_MODEL=True,
SCHEDULER_FACTOR=0.5, SCHEDULER_PATIENCE=0, TASK=2, VOCAB_SIZE=90235,
WEIGHT_DECAY=0.0005)
```

## 8 Sanity Check

- Check the loss without any training. For Cross entropy the expected value will be  $\log(\text{number of classes})$

```
[61]: device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
```

```
[62]: device
```

```
[62]: device(type='cuda', index=0)
```

```
[63]: for input_, targets, offsets in train_loader:

    # move inputs and outputs to GPUs
    input_ = input_.to(device)
    targets = targets.to(device)
    offsets = offsets.to(device)
    model_multilabel.eval()
    # Forward pass
    output = model_multilabel(input_, offsets)
    loss = loss_function(output, targets.float())
    print(f'Actual loss: {loss}')
    break

print(f'Expected Theoretical loss: {np.log(2)}')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
```



```
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()

Actual loss: 0.6931886672973633
Expected Theoretical loss: 0.6931471805599453
```

## 9 Training Model

```
[64]: wandb.watch(model_multilabel, log = 'all', log_freq=25, log_graph=True)
```

wandb: logging graph, to disable use `wandb.watch(log\_graph=False)`

```
[64]: [<wandb.wandb_torch.TorchGraph at 0x7fd39ebce290>]
```

```
[65]: import torch
torch.cuda.empty_cache()
```

```
[66]: # See live graphs in the notebook.
      #%%wandb
      # See live graphs in the notebook.
      #%%wandb
      batch_ct_train, batch_ct_valid = 0, 0
      train_loss_history, valid_loss_history = train_loop(train_loader,
                                                           valid_loader,

      ↪          model_multilabel,
      ↪
      ↪          optimizer,
      ↪
      ↪          loss_function,
      ↪
      ↪          wandb.config.EPOCHS,
      ↪
      ↪          wandb.config.DEVICE,
      ↪
      ↪          wandb.config.PATIENCE,
      ↪
      ↪          wandb.config.EARLY_STOPPING,
      ↪
      ↪          wandb.config.FILE_MODEL,
      ↪
      ↪          wandb.config.SAVE_BEST_MODEL)
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before

```

converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Validation loss has decreased (inf --> 0.308549). Saving Model...
Epoch : 1 / 10

```

Time to complete 1 is 0:01:21.026077

Learning rate: 0.02

Train Loss: 0.2139

Valid Loss: 0.3085

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
```

Validation loss has decreased (0.308549 --> 0.178167). Saving model...

Epoch : 2 / 10

Time to complete 2 is 0:01:25.301858

Learning rate: 0.02

Train Loss: 0.1709

Valid Loss: 0.1782

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
```

```

../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Validation loss has decreased (0.178167 --> 0.175621). Saving model...
Epoch : 3 / 10
Time to complete 3 is 0:01:23.963741
Learning rate: 0.02
Train Loss:  0.1617
Valid Loss:  0.1756

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please

```

```

consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()

Validation loss has decreased (0.175621 --> 0.153460). Saving model...
Epoch : 4 / 10
Time to complete 4 is 0:01:25.236507
Learning rate: 0.02
Train Loss: 0.1475
Valid Loss: 0.1535

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Epoch 00005: reducing learning rate of group 0 to 1.0000e-02.
Early stopping counter: 1 out of 10
Epoch : 5 / 10
Time to complete 5 is 0:01:26.931339
Learning rate: 0.01
Train Loss: 0.1392
Valid Loss: 0.1562

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before

```

```

converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Validation loss has decreased (0.153460 --> 0.133675). Saving model...
Epoch : 6 / 10
Time to complete 6 is 0:01:22.641608
Learning rate: 0.01
Train Loss: 0.1236
Valid Loss: 0.1337

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:

```



```

Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

```

```
Epoch 00007: reducing learning rate of group 0 to 5.0000e-03.  
Early stoping counter: 1 out of 10  
Epoch : 7 / 10  
Time to complete 7 is 0:01:24.845663  
Learning rate: 0.005  
Train Loss: 0.1229  
Valid Loss: 0.1378
```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

```

```

converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Validation loss has decreased (0.133675 --> 0.125764). Saving model...
Epoch : 8 / 10
Time to complete 8 is 0:01:21.814651
Learning rate: 0.005
Train Loss: 0.1071
Valid Loss: 0.1258

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:

```

```

Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()

Epoch 00009: reducing learning rate of group 0 to 2.5000e-03.
Early stopping counter: 1 out of 10
Epoch : 9 / 10
Time to complete 9 is 0:01:26.815689
Learning rate: 0.0025
Train Loss: 0.1048
Valid Loss: 0.1366

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at

```

```

../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Validation loss has decreased (0.125764 --> 0.119679). Saving model...
Epoch : 10 / 10
Time to complete 10 is 0:01:21.955316
Learning rate: 0.0025
Train Loss: 0.0927
Valid Loss: 0.1197

```

## 10 Get Accuracy, Predictions

```
[67]: device
```

```
[67]: device(type='cuda', index=0)
```

```
[68]: model_nn = MLPCustom(wandb.config.EMBED_DIM,
                           wandb.config.VOCAB_SIZE,
                           wandb.config.HIDDEN_SIZES_LIST,
                           wandb.config.DPROB_LIST,
                           wandb.config.OUTPUT_DIM,
```

```

        wandb.config.NON_LINEARITY,
        wandb.config.BATCH_NORM,
        wandb.config.TASK,
        wandb.config.PRETRAINED_WEIGHTS_TENSOR)
model_nn.to(wandb.config.DEVICE)
model_nn.load_state_dict(torch.load(wandb.config.FILE_MODEL))

```

[68]: <All keys matched successfully>

```

[69]: # Get the prediction and labels
y_train, y_predicted_train = get_pred(train_loader, model_nn)
y_valid, y_predicted_valid = get_pred(valid_loader, model_nn)
y_test, y_predicted_test = get_pred(test_loader, model_nn)

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

```

```

converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

```

```
[70]: pip install torchmetrics
```

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting torchmetrics
  Downloading torchmetrics-0.10.3-py3-none-any.whl (529 kB)
    |                               | 529 kB 15.4 MB/s
Requirement already satisfied: typing-extensions in

```

```
/usr/local/lib/python3.7/dist-packages (from torchmetrics) (4.1.1)
Requirement already satisfied: numpy>=1.17.2 in /usr/local/lib/python3.7/dist-
packages (from torchmetrics) (1.21.6)
Requirement already satisfied: torch>=1.3.1 in /usr/local/lib/python3.7/dist-
packages (from torchmetrics) (1.12.1+cu113)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-
packages (from torchmetrics) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in
/usr/local/lib/python3.7/dist-packages (from packaging->torchmetrics) (3.0.9)
Installing collected packages: torchmetrics
Successfully installed torchmetrics-0.10.3
```

```
[71]: from torchmetrics import F1Score
```

```
[72]: f1score = F1Score(num_classes=10, mdmc_average= 'global').to(device)
```

```
[73]: train_f1_score = f1score( y_predicted_train, y_train.long())
```

```
[74]: train_f1_score
```

```
[74]: tensor(0.9268, device='cuda:0')
```

```
[75]: # convert these to numpy array
y_train, y_predicted_train = y_train.cpu().numpy(), y_predicted_train.cpu().
    ↪numpy()
y_valid, y_predicted_valid = y_valid.cpu().numpy(), y_predicted_valid.cpu().
    ↪numpy()
y_test, y_predicted_test = y_test.cpu().numpy(), y_predicted_test.cpu().numpy()
```

```
[76]: from sklearn.metrics import f1_score
```

```
[77]: f1_score_train = f1_score(y_train, y_predicted_train, average = 'micro')
f1_score_valid = f1_score(y_valid, y_predicted_valid, average = 'micro')
f1_score_test = f1_score(y_test, y_predicted_test, average = 'micro')
```

```
[78]: # Print Accuracy based on saved Model
print('f1_score_train', f1_score_train)
print('f1_score_valid', f1_score_valid)
print('f1_score_test', f1_score_test)
```

```
f1_score_train 0.9267560994212012
f1_score_valid 0.8986018454617489
f1_score_test 0.8956792327704942
```

```
[79]: wandb.log({'Train f1 score': f1_score_train})
wandb.log({'Valid f1 score': f1_score_valid})
wandb.log({'Test f1 score': f1_score_test})
```



```
[80]: wandb.finish()
```

<IPython.core.display.HTML object>

VBox(children=(Label(value='0.003 MB of 0.003 MB uploaded (0.000 MB deduped)\r'), FloatProgress(),

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

## 11 TASK 4

### 11.1 Import libraries

```
[81]: # Importing the necessary libraries
import numpy as np
import pandas as pd

from pathlib import Path

from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import plot_confusion_matrix
import matplotlib.pyplot as plt
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from sklearn.multiclass import OneVsRestClassifier
from nltk.corpus import stopwords
from sklearn.svm import LinearSVC
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
import seaborn as sns

import custom_preprocessor as cp
from plot_learning_curve import plot_learning_curve as plc

import gensim
from gensim.models import KeyedVectors
```

```
import sys
import joblib
```

## 11.2 Gensim vectorizer

```
[82]: from sklearn.base import BaseEstimator, TransformerMixin
from collections import Counter
import numpy as np

class GensimVectorizer(BaseEstimator,TransformerMixin):
    np.random.seed(0)
    def __init__(self,pretrained_vectors,unk_norm_init=False):
        # load in pre-trained word vectors
        self.pretrained_vectors= pretrained_vectors
        self.vec_size= self.pretrained_vectors.vector_size
        self.unk_norm_init = unk_norm_init
        self.pretrained_vectors_subset = {}
        self.words_not_in_pretrained = []
        self.count_missing = 0
        self.percent_missing = 0

    def fit(self, X,y=None):
        '''
        Gets the subset of pretrained vectors which are present in vocab
        X : training sentences
        '''
        counter = Counter()

        for sent in X:
            counter.update(sent.split())
        for token in counter:
            try:
                self.pretrained_vectors_subset[token] = self.pretrained_vectors.
→get_vector(token, norm=True)
            except:
                self.words_not_in_pretrained.append(token)

        ### save so that you can access this after you fit the vectorizer
        self.count_missing = len(self.words_not_in_pretrained )
        self.percent_missing = self.count_missing / len(counter)
        return self

    def transform(self,X,y=None):
        X_vector = np.zeros((len(X), self.vec_size))
```

```

for i, sent in enumerate(X):
    sent_vector = np.zeros(self.vec_size)
    n=0
    tokens = sent.split()
    for word in tokens:
        if word in self.pretrained_vectors_subset.keys():
            word_vector=self.pretrained_vectors_subset[word]
            sent_vector+= word_vector
            n+= 1
        else:
            if self.unk_norm_init :
                word_vector = np.random.normal(size= self.vec_size)
                sent_vector+= word_vector
                n+= 1
    if n>0:
        X_vector[i] = sent_vector/n
return X_vector

```

## 12 Classification Pipeline

```

[83]: pretrained_vectors = KeyedVectors.load('/content/drive/MyDrive/NLP/
↳models/model_cbow.bin')

```

```

[84]: y_train

```

```

[84]: array([[1., 1., 0., ..., 0., 0., 0.],
            [0., 0., 0., ..., 0., 0., 0.],
            [0., 0., 0., ..., 0., 0., 1.],
            ...,
            [0., 0., 0., ..., 0., 0., 0.],
            [0., 1., 0., ..., 0., 0., 0.],
            [1., 0., 0., ..., 0., 0., 1.]], dtype=float32)

```

```

[85]: # OneVsRest strategy can be used for multi-label learning,
# where a classifier is used to predict multiple labels for instance.
# Naive Bayes supports multi-class, but we are in a multi-label scenario,
# therefore, we can wrap Naive Bayes in the OneVsRestClassifier
# Ref: https://towardsdatascience.com/
↳multi-label-text-classification-with-scikit-learn-30714b7819c5

```

```

[86]: # We will wrap Logistic Regression in OneVsRestClassifier
# (since we decompose it into multiple independent binary classification
↳problems)
# Reference: https://towardsdatascience.com/
↳journey-to-the-center-of-multi-label-classification-384c40229bff

```

```
[87]: categories = ['0','1','2','3','4','5','6','7','8','9']
```

```
[105]: from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score
from sklearn.multiclass import OneVsRestClassifier
# Using pipeline for applying logistic regression and one vs rest classifier
pipeline = Pipeline([
    ('vectorizer', GensimVectorizer(pretrained_vectors)),
    ('clf', OneVsRestClassifier(LogisticRegression(solver='sag'),
    ↪n_jobs=-1)),
])
for category in categories:
    print('**Processing {} comments...**'.format(category))

    # Training logistic regression model on train data
    pipeline.fit(X_train, y_train)

    # calculating test accuracy
    prediction = pipeline.predict(X_test)
    print('Test accuracy is {}'.format(accuracy_score(y_test, prediction)))
    print("\n")
```

```
**Processing 0 comments...**
Test accuracy is 0.3777145266708834
```

```
**Processing 1 comments...**
Test accuracy is 0.3776091081593928
```

```
**Processing 2 comments...**
Test accuracy is 0.37781994518237405
```

```
**Processing 3 comments...**
Test accuracy is 0.3777145266708834
```

```
**Processing 4 comments...**
Test accuracy is 0.3777145266708834
```

```
**Processing 5 comments...**
Test accuracy is 0.3777145266708834
```

```
**Processing 6 comments...**  
Test accuracy is 0.3777145266708834
```

```
**Processing 7 comments...**  
Test accuracy is 0.3777145266708834
```

```
**Processing 8 comments...**  
Test accuracy is 0.3777145266708834
```

```
**Processing 9 comments...**  
Test accuracy is 0.3777145266708834
```

```
[89]: # We will use BinaryRelevance  
# An ensemble of single-label binary classifiers is trained, one for each class.  
↪  
# Each classifier predicts either the membership or the non-membership of one  
↪class.  
# The union of all classes that were predicted is taken as the multi-label  
↪output  
# Reference: https://towardsdatascience.com/  
↪journey-to-the-center-of-multi-label-classification-384c40229bff
```

```
[90]: !pip install scikit-multilearn  
import skmultilearn
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/  
Collecting scikit-multilearn  
  Downloading scikit_multilearn-0.2.0-py3-none-any.whl (89 kB)  
      |                               | 89 kB 5.2 MB/s  
Installing collected packages: scikit-multilearn  
Successfully installed scikit-multilearn-0.2.0
```

```
[91]: X_train.dtype
```

```
[91]: dtype('O')
```

```
[92]: # using binary relevance  
from skmultilearn.problem_transform import BinaryRelevance  
from sklearn.naive_bayes import GaussianNB  
# Using pipeline for applying logistic regression and one vs rest classifier  
pipeline = Pipeline([
```

```

        ('vectorizer', GensimVectorizer(pretrained_vectors)),
        ('clf', BinaryRelevance(GaussianNB()))
    ])
for category in categories:
    print('**Processing {} comments...**'.format(category))

    # Training logistic regression model on train data
    pipeline.fit(X_train, y_train)

    # calculating test accuracy
    prediction = pipeline.predict(X_test)
    print('Test accuracy is {}'.format(accuracy_score(y_test, prediction)))
    print("\n")

```

```

**Processing 0 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 1 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 2 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 3 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 4 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 5 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 6 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 7 comments...**
Test accuracy is 0.2787265443811933

```

```

**Processing 8 comments...**

```

Test accuracy is 0.2787265443811933

**\*\*Processing 9 comments...\*\***

Test accuracy is 0.2787265443811933

```
[93]: # We will use ClassifierChain  
# A chain of binary classifiers C0, C1, . . . , Cn is constructed,  
# where a classifier Ci uses the predictions of all the classifier Cj , where j  
    ↪ < i  
# The total number of classifiers needed for this approach is equal to the  
    ↪ number of classes,  
# but the training of the classifiers is more involved  
# Reference: https://towardsdatascience.com/  
    ↪ journey-to-the-center-of-multi-label-classification-384c40229bff
```

```
[94]: from skmultilearn.problem_transform import ClassifierChain  
from sklearn.linear_model import LogisticRegression  
# Using pipeline for applying logistic regression and one vs rest classifier  
pipeline = Pipeline([  
    ('vectorizer',GensimVectorizer(pretrained_vectors)),  
    ('clf', ClassifierChain(LogisticRegression()))  
])  
for category in categories:  
    print('**Processing {} comments...**'.format(category))  
  
    # Training logistic regression model on train data  
    pipeline.fit(X_train, y_train)  
  
    # calculating test accuracy  
    prediction = pipeline.predict(X_test)  
    print('Test accuracy is {}'.format(accuracy_score(y_test, prediction)))  
    print("\n")
```

**\*\*Processing 0 comments...\*\***

Test accuracy is 0.4174573055028463

**\*\*Processing 1 comments...\*\***

Test accuracy is 0.4174573055028463

**\*\*Processing 2 comments...\*\***

Test accuracy is 0.4174573055028463

```
**Processing 3 comments...**  
Test accuracy is 0.4174573055028463
```

```
**Processing 4 comments...**  
Test accuracy is 0.4174573055028463
```

```
**Processing 5 comments...**  
Test accuracy is 0.4174573055028463
```

```
**Processing 6 comments...**  
Test accuracy is 0.4174573055028463
```

```
**Processing 7 comments...**  
Test accuracy is 0.4174573055028463
```

```
**Processing 8 comments...**  
Test accuracy is 0.4174573055028463
```

```
**Processing 9 comments...**  
Test accuracy is 0.4174573055028463
```

## 13 TASK 5

## 14 Weight Matrix of Pretrained Weights

```
[95]: pretrained_vectors = KeyedVectors.load('/content/drive/MyDrive/NLP,  
↳models/model_cbow.bin')
```

```
[96]: vector = pretrained_vectors.get_vector('office', norm=True)  
vector.shape
```

```
[96]: (150,)
```

```
[97]: embedding_dim = 150  
test_weights = np.zeros((2, embedding_dim))
```

```
[98]: test_weights[0] = pretrained_vectors.get_vector('office', norm=True)
```



```
[99]: test_weights[1] = np.random.normal(size=(embedding_dim, ))
```

```
[100]: test_weights
```

```
[100]: array([[ -6.12220392e-02,  -6.43143728e-02,  -1.47253862e-02,
        -1.68681908e-02,  -6.82770163e-02,  -5.03983870e-02,
         9.00510885e-03,   1.10479690e-01,  -3.85514833e-02,
        -4.42132428e-02,   1.83668256e-01,   1.81970596e-02,
        -6.72796043e-03,   2.67487913e-02,   7.99450949e-02,
        -1.51483670e-01,   5.40428795e-02,   3.35344672e-02,
        -4.83859368e-02,   9.37992260e-02,  -2.63110530e-02,
        -1.65316924e-01,   8.53660330e-02,  -4.42277566e-02,
        -6.09044060e-02,  -2.16226317e-02,  -1.53136542e-02,
         5.85562922e-02,   1.46070585e-01,  -8.45404193e-02,
         4.10615429e-02,  -6.62873238e-02,   6.91241622e-02,
        -5.02445959e-02,  -5.96107729e-02,  -1.06516331e-01,
         1.34776114e-02,  -5.81563124e-03,   3.57534960e-02,
         4.88940924e-02,  -9.89808887e-02,   9.90034342e-02,
         1.09738089e-01,   7.10285082e-02,  -4.50888872e-02,
         3.54727288e-03,  -9.30363834e-02,  -8.72039348e-02,
         3.26663516e-02,  -1.55520171e-01,   2.53420491e-02,
         8.86818245e-02,  -4.42792401e-02,  -5.02732955e-02,
        -2.06673536e-02,   3.97099145e-02,   3.80135030e-02,
         1.82009161e-01,  -1.30161375e-01,  -1.44659253e-02,
         4.58315723e-02,   8.09352659e-03,   2.05521611e-03,
         2.65751276e-02,  -5.52023090e-02,  -7.20347241e-02,
         1.21202832e-02,  -1.07882574e-01,   3.65291419e-03,
        -8.22809041e-02,   6.98235929e-02,   6.80745766e-02,
         3.48363370e-02,   9.29150507e-02,  -1.11451879e-01,
        -1.32282466e-01,  -2.00030822e-02,  -1.83615368e-02,
         8.80272985e-02,   1.28395915e-01,  -7.01950490e-02,
        -4.89731021e-02,   1.06386006e-01,  -6.18734257e-03,
        -1.16648443e-01,  -1.60571877e-02,  -8.80076587e-02,
        -6.44770265e-02,   8.61187056e-02,  -3.56374197e-02,
         1.98511537e-02,  -1.76975548e-01,   3.22094709e-02,
        -5.23201302e-02,   3.12646963e-02,  -5.53193390e-02,
         3.93999405e-02,  -3.79181057e-02,  -1.86040699e-02,
         8.55109468e-02,   3.97456922e-02,  -4.27951403e-02,
         1.67984620e-01,  -5.31589389e-02,   1.60806924e-01,
         5.19003458e-02,  -7.83144012e-02,   1.16425797e-01,
        -1.45005554e-01,  -6.61792234e-02,   5.15389182e-02,
         7.22263157e-02,  -9.86291468e-02,  -2.70440634e-02,
         9.12757665e-02,   1.49567872e-02,   6.35119081e-02,
        -2.95039807e-02,   1.56396270e-01,  -1.63303185e-02,
         6.26122132e-02,   1.01852715e-01,  -1.34266704e-01,
        -2.86244676e-02,  -9.68819018e-03,  -5.19974418e-02,
        -9.87500623e-02,   1.18297875e-01,  -7.91054070e-02,
```

-1.64527327e-01, -5.23377471e-02, 7.96608031e-02,  
 -6.17654845e-02, 8.22246447e-02, 1.54829293e-01,  
 3.48234810e-02, -1.76987380e-01, -3.83714326e-02,  
 -1.12718053e-01, 4.54049446e-02, -4.35751453e-02,  
 2.98597421e-02, 1.17147818e-01, -2.07625419e-01,  
 4.29562256e-02, -4.19769548e-02, 9.27577019e-02,  
 5.04710013e-03, -7.68736526e-02, 3.94994467e-02],  
 [ 1.76405235e+00, 4.00157208e-01, 9.78737984e-01,  
 2.24089320e+00, 1.86755799e+00, -9.77277880e-01,  
 9.50088418e-01, -1.51357208e-01, -1.03218852e-01,  
 4.10598502e-01, 1.44043571e-01, 1.45427351e+00,  
 7.61037725e-01, 1.21675016e-01, 4.43863233e-01,  
 3.33674327e-01, 1.49407907e+00, -2.05158264e-01,  
 3.13067702e-01, -8.54095739e-01, -2.55298982e+00,  
 6.53618595e-01, 8.64436199e-01, -7.42165020e-01,  
 2.26975462e+00, -1.45436567e+00, 4.57585173e-02,  
 -1.87183850e-01, 1.53277921e+00, 1.46935877e+00,  
 1.54947426e-01, 3.78162520e-01, -8.87785748e-01,  
 -1.98079647e+00, -3.47912149e-01, 1.56348969e-01,  
 1.23029068e+00, 1.20237985e+00, -3.87326817e-01,  
 -3.02302751e-01, -1.04855297e+00, -1.42001794e+00,  
 -1.70627019e+00, 1.95077540e+00, -5.09652182e-01,  
 -4.38074302e-01, -1.25279536e+00, 7.77490356e-01,  
 -1.61389785e+00, -2.12740280e-01, -8.95466561e-01,  
 3.86902498e-01, -5.10805138e-01, -1.18063218e+00,  
 -2.81822283e-02, 4.28331871e-01, 6.65172224e-02,  
 3.02471898e-01, -6.34322094e-01, -3.62741166e-01,  
 -6.72460448e-01, -3.59553162e-01, -8.13146282e-01,  
 -1.72628260e+00, 1.77426142e-01, -4.01780936e-01,  
 -1.63019835e+00, 4.62782256e-01, -9.07298364e-01,  
 5.19453958e-02, 7.29090562e-01, 1.28982911e-01,  
 1.13940068e+00, -1.23482582e+00, 4.02341641e-01,  
 -6.84810091e-01, -8.70797149e-01, -5.78849665e-01,  
 -3.11552532e-01, 5.61653422e-02, -1.16514984e+00,  
 9.00826487e-01, 4.65662440e-01, -1.53624369e+00,  
 1.48825219e+00, 1.89588918e+00, 1.17877957e+00,  
 -1.79924836e-01, -1.07075262e+00, 1.05445173e+00,  
 -4.03176947e-01, 1.22244507e+00, 2.08274978e-01,  
 9.76639036e-01, 3.56366397e-01, 7.06573168e-01,  
 1.05000207e-02, 1.78587049e+00, 1.26912093e-01,  
 4.01989363e-01, 1.88315070e+00, -1.34775906e+00,  
 -1.27048500e+00, 9.69396708e-01, -1.17312341e+00,  
 1.94362119e+00, -4.13618981e-01, -7.47454811e-01,  
 1.92294203e+00, 1.48051479e+00, 1.86755896e+00,  
 9.06044658e-01, -8.61225685e-01, 1.91006495e+00,  
 -2.68003371e-01, 8.02456396e-01, 9.47251968e-01,  
 -1.55010093e-01, 6.14079370e-01, 9.22206672e-01,

```

3.76425531e-01, -1.09940079e+00, 2.98238174e-01,
1.32638590e+00, -6.94567860e-01, -1.49634540e-01,
-4.35153552e-01, 1.84926373e+00, 6.72294757e-01,
4.07461836e-01, -7.69916074e-01, 5.39249191e-01,
-6.74332661e-01, 3.18305583e-02, -6.35846078e-01,
6.76433295e-01, 5.76590817e-01, -2.08298756e-01,
3.96006713e-01, -1.09306151e+00, -1.49125759e+00,
4.39391701e-01, 1.66673495e-01, 6.35031437e-01,
2.38314477e+00, 9.44479487e-01, -9.12822225e-01,
1.11701629e+00, -1.31590741e+00, -4.61584605e-01]])

```

```
[101]: len(multilabel_vocab)
```

```
[101]: 90235
```

```

[102]: embedding_dim = 150
pretrained_weights = np.zeros((len(multilabel_vocab), embedding_dim))
words_found = 0
words_not_found = 0

for i, word in enumerate(multilabel_vocab.get_itos()):
    try:
        pretrained_weights[i] = pretrained_vectors.get_vector(word, norm=True)
        words_found += 1
    except KeyError:
        words_not_found += 1
        pretrained_weights[i] = np.random.normal(size=(embedding_dim, ))

```

```
[103]: words_found
```

```
[103]: 11487
```

```
[104]: words_not_found
```

```
[104]: 78748
```

## 15 Meta Data

```

[106]: hyperparameters = SimpleNamespace(
    EMBED_DIM = 150,
    VOCAB_SIZE = len(multilabel_vocab),
    OUTPUT_DIM = 10,
    HIDDEN_SIZES_LIST = [500,200], # 100 layers of size 200 [200]*100
    DPROB_LIST = [0,0],
    NON_LINEARITY= nn.SELU(),

```

```

PRETRAINED_WEIGHTS_TENSOR = torch.tensor(pretrained_weights).float(),
BATCH_NORM = True,
EPOCHS = 10,
TASK = 5,
BATCH_SIZE = 256,
LEARNING_RATE = 0.02,
DATASET="MultiLabel",
ARCHITECTUREe="2_hidden_layers",
LOG_INTERVAL = 25,
LOG_BATCH = True,
FILE_MODEL = save_model_folder/' ' .pt',
GRAD_CLIPPING = False,
MAX_NORM = 0,
MOMENTUM = 0,
PATIENCE = 10,
EARLY_STOPPING = True,
SCHEDULER_FACTOR = 0.5,
SCHEDULER_PATIENCE = 0,
WEIGHT_DECAY = 0.0005,
SAVE_BEST_MODEL = True,
DEVICE = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
)

```

```
[107]: torch.cuda.is_available()
```

```
[107]: True
```

## 16 Data Loaders, Loss Function, Optimizer

```

[108]: # Initialize a new project
import random
wandb.init(name = 'task2', project = 'NLP_ ', config = hyperparameters)

```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
[108]: <wandb.sdk.wandb_run.Run at 0x7fd091e872d0>
```

```

[109]: wandb.config = hyperparameters
wandb.config

```

```
[109]: namespace(ARCHITECTURE='2_hidden_layers', BATCH_NORM=True, BATCH_SIZE=256,
  DATASET='MultiLabel', DEVICE=device(type='cuda', index=0), DPROB_LIST=[0, 0],
  EARLY_STOPPING=True, EMBED_DIM=150, EPOCHS=10, FILE_MODEL=PosixPath('/content/drive/MyDrive/NLP/'), GRAD_CLIPPING=False,
  HIDDEN_SIZES_LIST=[500, 200], LEARNING_RATE=0.02, LOG_BATCH=True,
  LOG_INTERVAL=25, MAX_NORM=0, MOMENTUM=0, NON_LINEARITY=SELU(), OUTPUT_DIM=10,
  PATIENCE=10, PRETRAINED_WEIGHTS_TENSOR=tensor([[-0.0682,  1.7133, -0.7448, ...,
  0.5830, -0.3994,  0.3701],
  [ 0.0588,  0.0114,  0.0397, ..., -0.1514, -0.0105, -0.0706],
  [-0.0205,  0.0141,  0.1818, ..., -0.1018,  0.0951,  0.0750],
  ...,
  [ 0.3919,  0.8843, -1.0461, ...,  0.9146, -1.9827, -0.7953],
  [-0.3745,  1.1140, -0.2199, ..., -1.4224,  0.8445,  1.4767],
  [ 0.8993, -0.2509, -1.6339, ..., -0.7607,  0.8491, -0.0463]]),
  SAVE_BEST_MODEL=True, SCHEDULER_FACTOR=0.5, SCHEDULER_PATIENCE=0, TASK=5,
  VOCAB_SIZE=90235, WEIGHT_DECAY=0.0005)
```

```
[110]: # Fix seed value
SEED = 2345
random.seed(SEED)
np.random.seed(SEED)
torch.manual_seed(SEED)
torch.cuda.manual_seed(SEED)
torch.backends.cudnn.deterministic = True

# Data Loader
train_loader = torch.utils.data.DataLoader(trainset, batch_size=wandb.config.
  ↳BATCH_SIZE, shuffle = True,
                                     collate_fn=collate_batch,
  ↳num_workers = 4)
valid_loader = torch.utils.data.DataLoader(validset, batch_size=wandb.config.
  ↳BATCH_SIZE, shuffle = False,
                                     collate_fn=collate_batch,
  ↳num_workers = 4)
test_loader = torch.utils.data.DataLoader(testset, batch_size=wandb.config.
  ↳BATCH_SIZE, shuffle = False,
                                     collate_fn=collate_batch,
  ↳num_workers = 4)

# cross entropy loss function
loss_function = nn.BCEWithLogitsLoss()

# model
model_multilabel = MLPCustom(wandb.config.EMBED_DIM,
                             wandb.config.VOCAB_SIZE,
                             wandb.config.HIDDEN_SIZES_LIST,
                             wandb.config.DPROB_LIST,
```

```

        wandb.config.OUTPUT_DIM,
        wandb.config.NON_LINEARITY,
        wandb.config.BATCH_NORM,
        wandb.config.TASK,
        wandb.config.PRETRAINED_WEIGHTS_TENSOR)

model_multilabel.to(wandb.config.DEVICE)

def init_weights(m):
    if type(m) == nn.Linear:
        torch.nn.init.kaiming_normal_(m.weight)
        torch.nn.init.zeros_(m.bias)

# apply initialization recursively to all modules
# model_multilabel.apply(init_weights)

# Initialize stochastic gradient descent optimizer
optimizer = torch.optim.Adam(model_multilabel.parameters(),
                              lr = wandb.config.LEARNING_RATE,
                              weight_decay=wandb.config.WEIGHT_DECAY)

# wandb.config.OPTIMIZER = optimizer

scheduler = ReduceLROnPlateau(optimizer, mode='min', factor= wandb.config.
    ↪ SCHEDULER_FACTOR,
                              patience=wandb.config.SCHEDULER_PATIENCE,
    ↪ verbose=True)

# scheduler = StepLR(optimizer, gamma=0.4, step_size=1, verbose=True)

```

```

/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py:566:
UserWarning: This DataLoader will create 4 worker processes in total. Our
suggested max number of worker in current system is 2, which is smaller than
what this DataLoader is going to create. Please be aware that excessive worker
creation might get DataLoader running slow or even freeze, lower the worker
number to avoid potential slowness/freeze if necessary.
  cpuset_checked))

```

```
[111]: wandb.config.DEVICE
```

```
[111]: device(type='cuda', index=0)
```

```
[112]: wandb.config
```

```
[112]: namespace(ARCHITECTURE='2_hidden_layers', BATCH_NORM=True, BATCH_SIZE=256,
DATASET='MultiLabel', DEVICE=device(type='cuda', index=0), DPROB_LIST=[0, 0],
EARLY_STOPPING=True, EMBED_DIM=150, EPOCHS=10, FILE_MODEL=PosixPath('/content/dr

```

```

ive/MyDrive/NLP/I_...pt'), GRAD_CLIPPING=False,
HIDDEN_SIZES_LIST=[500, 200], LEARNING_RATE=0.02, LOG_BATCH=True,
LOG_INTERVAL=25, MAX_NORM=0, MOMENTUM=0, NON_LINEARITY=SELU(), OUTPUT_DIM=10,
PATIENCE=10, PRETRAINED_WEIGHTS_TENSOR=tensor([[ -0.0682,  1.7133, -0.7448, ...,
0.5830, -0.3994,  0.3701],
[ 0.0588,  0.0114,  0.0397, ..., -0.1514, -0.0105, -0.0706],
[ -0.0205,  0.0141,  0.1818, ..., -0.1018,  0.0951,  0.0750],
...,
[ 0.3919,  0.8843, -1.0461, ...,  0.9146, -1.9827, -0.7953],
[ -0.3745,  1.1140, -0.2199, ..., -1.4224,  0.8445,  1.4767],
[ 0.8993, -0.2509, -1.6339, ..., -0.7607,  0.8491, -0.0463]]),
SAVE_BEST_MODEL=True, SCHEDULER_FACTOR=0.5, SCHEDULER_PATIENCE=0, TASK=5,
VOCAB_SIZE=90235, WEIGHT_DECAY=0.0005)

```

## 17 Sanity Check

- Check the loss without any training. For Cross entropy the expected value will be  $\log(\text{number of classes})$

```
[113]: device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
```

```
[114]: device
```

```
[114]: device(type='cuda', index=0)
```

```
[121]: for input_, targets, offsets in train_loader:

    # move inputs and outputs to GPUs
    input_ = input_.to(device)
    targets = targets.to(device)
    offsets = offsets.to(device)
    model_multilabel.eval()

    # Forward pass
    output = model_multilabel(input_, offsets)
    loss = loss_function(output, targets.float())
    print(f'Actual loss: {loss}')
    break

print(f'Expected Theoretical loss: {np.log(2)}')
```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Actual loss: 0.6939117312431335
Expected Theoretical loss: 0.6931471805599453

```

## 18 Training Model

```
[122]: wandb.watch(model_multilabel, log = 'all', log_freq=25, log_graph=True)
```

```
wandb: logging graph, to disable use `wandb.watch(log_graph=False)`
```

```
[122]: [<wandb.wandb_torch.TorchGraph at 0x7fd091a37910>]
```

```

[123]: # See live graphs in the notebook.
      %%wandb
      # See live graphs in the notebook.
      %%wandb
      batch_ct_train, batch_ct_valid = 0, 0
      train_loss_history, valid_loss_history = train_loop(train_loader,
                                                         valid_loader,

      ↪          model_multilabel,
      ↪
      ↪          optimizer,
      ↪
      ↪          loss_function,
      ↪
      ↪          wandb.config.EPOCHS,

```



```

→ wandb.config.DEVICE,
→ wandb.config.PATIENCE,
→ wandb.config.EARLY_STOPPING,
→ wandb.config.FILE_MODEL,
→ wandb.config.SAVE_BEST_MODEL)

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor_new.cpp:201.)
```

```
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor_new.cpp:201.)
```

```
app.launch_new_instance()
```

Validation loss has decreased (inf --> 0.255687). Saving Model...

Epoch : 1 / 10

Time to complete 1 is 0:00:06.820830

Learning rate: 0.02

Train Loss: 0.3005

Valid Loss: 0.2557

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor_new.cpp:201.)
```

```
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor_new.cpp:201.)
```

```
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor_new.cpp:201.)
```

```
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor_new.cpp:201.)
```

```
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at
```

```

../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Validation loss has decreased (0.255687 --> 0.248233). Saving model...
Epoch : 2 / 10
Time to complete 2 is 0:00:06.691172
Learning rate: 0.02
Train Loss: 0.2476
Valid Loss: 0.2482

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please

```

```

consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Epoch 00003: reducing learning rate of group 0 to 1.0000e-02.
Early stopping counter: 1 out of 10
Epoch : 3 / 10
Time to complete 3 is 0:00:06.716062
Learning rate: 0.01
Train Loss: 0.2428
Valid Loss: 0.2996

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)

```

```

    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Validation loss has decreased (0.248233 --> 0.225125). Saving model...
Epoch : 4 / 10
Time to complete 4 is 0:00:06.758438
Learning rate: 0.01
Train Loss: 0.2367
Valid Loss: 0.2251

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before

```

```

converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Validation loss has decreased (0.225125 --> 0.225097). Saving model...
Epoch : 5 / 10

```

Time to complete 5 is 0:00:06.635655

Learning rate: 0.01

Train Loss: 0.2348

Valid Loss: 0.2251

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
```

Epoch 00006: reducing learning rate of group 0 to 5.0000e-03.

Early stoping counter: 1 out of 10

Epoch : 6 / 10

Time to complete 6 is 0:00:06.671038

Learning rate: 0.005

Train Loss: 0.2344

Valid Loss: 0.2304

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
```



```

converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Validation loss has decreased (0.225097 --> 0.224271). Saving model...
Epoch : 7 / 10
Time to complete 7 is 0:00:06.829370
Learning rate: 0.005
Train Loss: 0.2288
Valid Loss: 0.2243

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:

```

Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

Validation loss has decreased (0.224271 --> 0.218827). Saving model...

Epoch : 8 / 10

Time to complete 8 is 0:00:06.669182

Learning rate: 0.005

Train Loss: 0.2282

Valid Loss: 0.2188

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```

    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Epoch 00009: reducing learning rate of group 0 to 2.5000e-03.
Early stopping counter: 1 out of 10
Epoch : 9 / 10
Time to complete 9 is 0:00:06.809452
Learning rate: 0.0025
Train Loss: 0.2291
Valid Loss: 0.2423

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please

```

```

consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Epoch 00010: reducing learning rate of group 0 to 1.2500e-03.
Early stopping counter: 2 out of 10
Epoch : 10 / 10
Time to complete 10 is 0:00:06.893529
Learning rate: 0.00125
Train Loss: 0.2256
Valid Loss: 0.2197

```

## 19 Get Accuracy, Predictions

```
[124]: device
```

```
[124]: device(type='cuda', index=0)
```

```
[125]: model_nn = MLPCustom(wandb.config.EMBED_DIM,
                           wandb.config.VOCAB_SIZE,
                           wandb.config.HIDDEN_SIZES_LIST,
                           wandb.config.DPROB_LIST,
                           wandb.config.OUTPUT_DIM,
                           wandb.config.NON_LINEARITY,
                           wandb.config.BATCH_NORM,
                           wandb.config.TASK,
                           wandb.config.PRETRAINED_WEIGHTS_TENSOR)
model_nn.to(wandb.config.DEVICE)
model_nn.load_state_dict(torch.load(wandb.config.FILE_MODEL))
```

```
[125]: <All keys matched successfully>
```

```
[126]: # Get the prediction and labels
y_train, y_predicted_train = get_pred(train_loader, model_nn)
y_valid, y_predicted_valid = get_pred(valid_loader, model_nn)
y_test, y_predicted_test = get_pred(test_loader, model_nn)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
```

```

Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:

```

Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at ../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

```
[127]: f1score = F1Score(num_classes=10, mdc_average= 'global').to(device)
```

```
[128]: train_f1_score = f1score( y_predicted_train, y_train.long())
```

```
[129]: train_f1_score
```

```
[129]: tensor(0.7647, device='cuda:0')
```

```
[130]: # convert these to numpy array
y_train, y_predicted_train = y_train.cpu().numpy(), y_predicted_train.cpu().
    ↪numpy()
y_valid, y_predicted_valid = y_valid.cpu().numpy(), y_predicted_valid.cpu().
    ↪numpy()
y_test, y_predicted_test = y_test.cpu().numpy(), y_predicted_test.cpu().numpy()
```

```
[131]: from sklearn.metrics import f1_score
```

```
[132]: f1_score_train = f1_score(y_train, y_predicted_train, average = 'micro')
f1_score_valid = f1_score(y_valid, y_predicted_valid, average = 'micro')
f1_score_test = f1_score(y_test, y_predicted_test, average = 'micro')
```

```
[133]: # Print Accuracy based on saved Model
print('f1_score_train', f1_score_train)
print('f1_score_valid', f1_score_valid)
print('f1_score_test', f1_score_test)
```

```
f1_score_train 0.7647206483540128
f1_score_valid 0.7733948690104521
f1_score_test 0.7722659430122117
```

```
[134]: wandb.log({'Train f1 score': f1_score_train})
wandb.log({'Valid f1 score': f1_score_valid})
wandb.log({'Test f1 score': f1_score_test})
```

```
[135]: wandb.finish()
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

## 20 TASK 6

## 21 Meta Data

```
[136]: hyperparameters = SimpleNamespace(
    EMBED_DIM = 150,
    VOCAB_SIZE = len(multilabel_vocab),
    OUTPUT_DIM = 10,
    HIDDEN_SIZES_LIST = [500,200], # 100 layers of size 200 [200]*100
    DPROB_LIST = [0,0],
    NON_LINEARITY= nn.SELU(),
    PRETRAINED_WEIGHTS_TENSOR = torch.tensor(pretrained_weights).float(),
    BATCH_NORM = True,
    EPOCHS = 10,
    TASK = 6,
    BATCH_SIZE = 256,
    LEARNING_RATE = 0.02,
    DATASET="MultiLabel",
    ARCHITECTUREe="2_hidden_layers",
    LOG_INTERVAL = 25,
    LOG_BATCH = True,
    FILE_MODEL = save_model_folder/' '.pt',
    GRAD_CLIPPING = False,
    MAX_NORM = 0,
    MOMENTUM = 0,
    PATIENCE = 10,
    EARLY_STOPPING = True,
    SCHEDULER_FACTOR = 0.5,
    SCHEDULER_PATIENCE = 0,
    WEIGHT_DECAY = 0.0005,
    SAVE_BEST_MODEL = True,
    DEVICE = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
)
```

```
[137]: torch.cuda.is_available()
```

```
[137]: True
```



## 22 Data Loaders, Loss Function, Optimizer

```
[138]: # Initialize a new project
import random
wandb.init(name = 'task2', project = ' ', config = hyperparameters)
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
[138]: <wandb.sdk.wandb_run.Run at 0x7fd091b0e310>
```

```
[139]: wandb.config = hyperparameters
      wandb.config
```

```
[139]: namespace(ARCHITECTUREe='2_hidden_layers', BATCH_NORM=True, BATCH_SIZE=256,
        DATASET='MultiLabel', DEVICE=device(type='cuda', index=0), DPROB_LIST=[0, 0],
        EARLY_STOPPING=True, EMBED_DIM=150, EPOCHS=10, FILE_MODEL=PosixPath('/content/drive/MyDrive/NLP/
        _ _ _ .pt'), GRAD_CLIPPING=False,
        HIDDEN_SIZES_LIST=[500, 200], LEARNING_RATE=0.02, LOG_BATCH=True,
        LOG_INTERVAL=25, MAX_NORM=0, MOMENTUM=0, NON_LINEARITY=SELU(), OUTPUT_DIM=10,
        PATIENCE=10, PRETRAINED_WEIGHTS_TENSOR=tensor([[ -0.0682,  1.7133, -0.7448, ...,
        0.5830, -0.3994,  0.3701],
        [ 0.0588,  0.0114,  0.0397, ..., -0.1514, -0.0105, -0.0706],
        [-0.0205,  0.0141,  0.1818, ..., -0.1018,  0.0951,  0.0750],
        ...,
        [ 0.3919,  0.8843, -1.0461, ...,  0.9146, -1.9827, -0.7953],
        [-0.3745,  1.1140, -0.2199, ..., -1.4224,  0.8445,  1.4767],
        [ 0.8993, -0.2509, -1.6339, ..., -0.7607,  0.8491, -0.0463]])),
        SAVE_BEST_MODEL=True, SCHEDULER_FACTOR=0.5, SCHEDULER_PATIENCE=0, TASK=6,
        VOCAB_SIZE=90235, WEIGHT_DECAY=0.0005)
```

```
[140]: # Fix seed value
        SEED = 2345
        random.seed(SEED)
        np.random.seed(SEED)
        torch.manual_seed(SEED)
        torch.cuda.manual_seed(SEED)
        torch.backends.cudnn.deterministic = True

# Data Loader
        train_loader = torch.utils.data.DataLoader(trainset, batch_size=wandb.config.
            ↪BATCH_SIZE, shuffle = True,
```

```

collate_fn=collate_batch,
    ↪num_workers = 4)
valid_loader = torch.utils.data.DataLoader(validset, batch_size=wandb.config.
    ↪BATCH_SIZE, shuffle = False,

collate_fn=collate_batch,
    ↪num_workers = 4)
test_loader = torch.utils.data.DataLoader(testset, batch_size=wandb.config.
    ↪BATCH_SIZE, shuffle = False,

collate_fn=collate_batch,
    ↪num_workers = 4)

# cross entropy loss function
loss_function = nn.BCEWithLogitsLoss()

# model
model_multilabel = MLPCustom(wandb.config.EMBED_DIM,
                             wandb.config.VOCAB_SIZE,
                             wandb.config.HIDDEN_SIZES_LIST,
                             wandb.config.DPROB_LIST,
                             wandb.config.OUTPUT_DIM,
                             wandb.config.NON_LINEARITY,
                             wandb.config.BATCH_NORM,
                             wandb.config.TASK,
                             wandb.config.PRETRAINED_WEIGHTS_TENSOR)

model_multilabel.to(wandb.config.DEVICE)

def init_weights(m):
    if type(m) == nn.Linear:
        torch.nn.init.kaiming_normal_(m.weight)
        torch.nn.init.zeros_(m.bias)

# apply initialization recursively to all modules
# model_multilabel.apply(init_weights)

# Intialize stochastic gradient descent optimizer
optimizer = torch.optim.Adam(model_multilabel.parameters(),
                              lr = wandb.config.LEARNING_RATE,
                              weight_decay=wandb.config.WEIGHT_DECAY)

# wandb.config.OPTIMIZER = optimizer

scheduler = ReduceLROnPlateau(optimizer, mode='min', factor= wandb.config.
    ↪SCHEDULER_FACTOR,

                              patience=wandb.config.SCHEDULER_PATIENCE,
    ↪verbose=True)

```

```
#scheduler = StepLR(optimizer, gamma=0.4, step_size=1, verbose=True)
```

```
/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py:566:  
UserWarning: This DataLoader will create 4 worker processes in total. Our  
suggested max number of worker in current system is 2, which is smaller than  
what this DataLoader is going to create. Please be aware that excessive worker  
creation might get DataLoader running slow or even freeze, lower the worker  
number to avoid potential slowness/freeze if necessary.  
cpuset_checked))
```

```
[141]: wandb.config.DEVICE
```

```
[141]: device(type='cuda', index=0)
```

```
[142]: wandb.config
```

```
[142]: namespace(ARCHITECTUREe='2_hidden_layers', BATCH_NORM=True, BATCH_SIZE=256,  
DATASET='MultiLabel', DEVICE=device(type='cuda', index=0), DPROB_LIST=[0, 0],  
EARLY_STOPPING=True, EMBED_DIM=150, EPOCHS=10, FILE_MODEL=PosixPath('/content/dr  
ive/MyDrive/NLP/  
'), GRAD_CLIPPING=False,  
HIDDEN_SIZES_LIST=[500, 200], LEARNING_RATE=0.02, LOG_BATCH=True,  
LOG_INTERVAL=25, MAX_NORM=0, MOMENTUM=0, NON_LINEARITY=SELU(), OUTPUT_DIM=10,  
PATIENCE=10, PRETRAINED_WEIGHTS_TENSOR=tensor([[ -0.0682,  1.7133, -0.7448, ...,  
0.5830, -0.3994,  0.3701],  
[ 0.0588,  0.0114,  0.0397, ..., -0.1514, -0.0105, -0.0706],  
[-0.0205,  0.0141,  0.1818, ..., -0.1018,  0.0951,  0.0750],  
...,  
[ 0.3919,  0.8843, -1.0461, ...,  0.9146, -1.9827, -0.7953],  
[-0.3745,  1.1140, -0.2199, ..., -1.4224,  0.8445,  1.4767],  
[ 0.8993, -0.2509, -1.6339, ..., -0.7607,  0.8491, -0.0463]]),  
SAVE_BEST_MODEL=True, SCHEDULER_FACTOR=0.5, SCHEDULER_PATIENCE=0, TASK=6,  
VOCAB_SIZE=90235, WEIGHT_DECAY=0.0005)
```

## 23 Sanity Check

- Check the loss without any training. For Cross entropy the expected value will be  $\log(\text{number of classes})$

```
[143]: device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
```

```
[144]: device
```

```
[144]: device(type='cuda', index=0)
```

```
[145]: for input_, targets, offsets in train_loader:

    # move inputs and outputs to GPUs
    input_ = input_.to(device)
    targets = targets.to(device)
    offsets = offsets.to(device)
    model_multilabel.eval()
    # Forward pass
    output = model_multilabel(input_, offsets)
    loss = loss_function(output, targets.float())
    print(f'Actual loss: {loss}')
    break

print(f'Expected Theoretical loss: {np.log(2)}')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Actual loss: 0.6945945620536804
Expected Theoretical loss: 0.6931471805599453
```

## 24 Training Model

```
[146]: wandb.watch(model_multilabel, log = 'all', log_freq=25, log_graph=True)
```

wandb: logging graph, to disable use `wandb.watch(log\_graph=False)`

```
[146]: [<wandb.wandb_torch.TorchGraph at 0x7fd0915a4250>]
```

```
[147]: # See live graphs in the notebook.
# %%wandb
# See live graphs in the notebook.
# %%wandb
batch_ct_train, batch_ct_valid = 0, 0
train_loss_history, valid_loss_history = train_loop(train_loader,
                                                    valid_loader,

    ↪      model_multilabel,
    ↪
    ↪      optimizer,
    ↪
    ↪      loss_function,
    ↪
    ↪      wandb.config.EPOCHS,
    ↪
    ↪      wandb.config.DEVICE,
    ↪
    ↪      wandb.config.PATIENCE,
    ↪
    ↪      wandb.config.EARLY_STOPPING,
    ↪
    ↪      wandb.config.FILE_MODEL,
    ↪
    ↪      wandb.config.SAVE_BEST_MODEL)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
```

Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```
app.launch_new_instance()
```

Validation loss has decreased (inf --> 0.192825). Saving Model...

Epoch : 1 / 10

Time to complete 1 is 0:00:07.229786

Learning rate: 0.02

Train Loss: 0.2430

Valid Loss: 0.1928

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please consider converting the list to a single numpy.ndarray with numpy.array() before converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

```

    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
Validation loss has decreased (0.192825 --> 0.151609). Saving model...
Epoch : 2 / 10
Time to complete 2 is 0:00:07.118540
Learning rate: 0.02

```





```
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
```

Epoch 00003: reducing learning rate of group 0 to 1.0000e-02.

Early stoping counter: 1 out of 10

Epoch : 3 / 10

Time to complete 3 is 0:00:07.142046

Learning rate: 0.01

Train Loss: 0.1406

Valid Loss: 0.1651

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
app.launch_new_instance()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
```

```

    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Validation loss has decreased (0.151609 --> 0.126412). Saving model...
Epoch : 4 / 10
Time to complete 4 is 0:00:07.272406
Learning rate: 0.01
Train Loss:  0.1244
Valid Loss:  0.1264

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

```

```

converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Epoch 00005: reducing learning rate of group 0 to 5.0000e-03.
Early stopping counter: 1 out of 10
Epoch : 5 / 10
Time to complete 5 is 0:00:07.206813
Learning rate: 0.005
Train Loss: 0.1200
Valid Loss: 0.1342

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Validation loss has decreased (0.126412 --> 0.121033). Saving model...
Epoch : 6 / 10
Time to complete 6 is 0:00:07.386245
Learning rate: 0.005
Train Loss: 0.1044
Valid Loss: 0.1210

```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at

```

```

../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

Validation loss has decreased (0.121033 --> 0.119801). Saving model...
Epoch : 7 / 10
Time to complete 7 is 0:00:07.170740
Learning rate: 0.005
Train Loss: 0.1000
Valid Loss: 0.1198

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please

```

```

consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

```

Epoch 00008: reducing learning rate of group 0 to 2.5000e-03.

Early stoping counter: 1 out of 10  
Epoch : 8 / 10  
Time to complete 8 is 0:00:07.131865  
Learning rate: 0.0025  
Train Loss: 0.0976  
Valid Loss: 0.1288

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at  
../torch/csrc/utils/tensor\_new.cpp:201.)

app.launch\_new\_instance()

/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:16: UserWarning:  
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please  
consider converting the list to a single numpy.ndarray with numpy.array() before  
converting to a tensor. (Triggered internally at

```

../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Validation loss has decreased (0.119801 --> 0.115113). Saving model...
Epoch : 9 / 10
Time to complete 9 is 0:00:07.318225
Learning rate: 0.0025
Train Loss: 0.0858
Valid Loss: 0.1151

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

```



```

consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
    app.launch_new_instance()

Epoch 00010: reducing learning rate of group 0 to 1.2500e-03.
Early stopping counter: 1 out of 10
Epoch : 10 / 10
Time to complete 10 is 0:00:07.163806
Learning rate: 0.00125
Train Loss: 0.0805
Valid Loss: 0.1232

```

## 25 Get Accuracy, Predictions

```
[148]: device
```

```
[148]: device(type='cuda', index=0)
```

```

[149]: model_nn = MLPCustom(wandb.config.EMBED_DIM,
                             wandb.config.VOCAB_SIZE,
                             wandb.config.HIDDEN_SIZES_LIST,
                             wandb.config.DPROB_LIST,
                             wandb.config.OUTPUT_DIM,
                             wandb.config.NON_LINEARITY,
                             wandb.config.BATCH_NORM,
                             wandb.config.TASK,
                             wandb.config.PRETRAINED_WEIGHTS_TENSOR)
model_nn.to(wandb.config.DEVICE)
model_nn.load_state_dict(torch.load(wandb.config.FILE_MODEL))

```

```
[149]: <All keys matched successfully>
```

[150]: *# Get the prediction and labels*

```
y_train, y_predicted_train = get_pred(train_loader, model_nn)
y_valid, y_predicted_valid = get_pred(valid_loader, model_nn)
y_test, y_predicted_test = get_pred(test_loader, model_nn)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
```

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: UserWarning:
Creating a tensor from a list of numpy.ndarrays is extremely slow. Please
consider converting the list to a single numpy.ndarray with numpy.array() before
converting to a tensor. (Triggered internally at
../torch/csrc/utils/tensor_new.cpp:201.)
  app.launch_new_instance()

```

```
[151]: f1score = F1Score(num_classes=10, mdc_average= 'global').to(device)
```

```
[152]: train_f1_score = f1score( y_predicted_train, y_train.long())
```

```
[153]: train_f1_score
```

```
[153]: tensor(0.9351, device='cuda:0')
```

```
[154]: # convert these to numpy array
y_train, y_predicted_train = y_train.cpu().numpy(), y_predicted_train.cpu().
    ↪numpy()
y_valid, y_predicted_valid = y_valid.cpu().numpy(), y_predicted_valid.cpu().
    ↪numpy()
y_test, y_predicted_test = y_test.cpu().numpy(), y_predicted_test.cpu().numpy()
```

```
[155]: from sklearn.metrics import f1_score
```

```
[156]: f1_score_train = f1_score(y_train, y_predicted_train, average = 'micro')
f1_score_valid = f1_score(y_valid, y_predicted_valid, average = 'micro')
f1_score_test = f1_score(y_test, y_predicted_test, average = 'micro')
```

```
[157]: # Print Accuracy based on saved Model
print('f1_score_train', f1_score_train)
print('f1_score_valid', f1_score_valid)
print('f1_score_test', f1_score_test)
```

```
f1_score_train 0.9350741419034677
f1_score_valid 0.9010492086477148
f1_score_test 0.8955049323705888
```

```
[158]: wandb.log({'Train f1 score': f1_score_train})
wandb.log({'Valid f1 score': f1_score_valid})
wandb.log({'Test f1 score': f1_score_test})
```

```
[159]: wandb.finish()
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```