Harikrishna_Dev_HW_5

October 24, 2023

1 Setup environment

[1]:
```
from pathlib import Path
import sys

if 'google.colab' in str(get_ipython()):
    from google.colab import drive  # Import Google Drive mounting utility
    drive.mount('/content/drive')  # Mount Google Drive

    # REPLACE WITH YOUR FOLDER

    base_folder = Path('/content/drive/MyDrive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning')

    data_folder =  Path('/content')

    !pip install pytorch-lightning==2.0.9 -qq
    !pip install torchmetrics -U -qq
    !pip install fastdownload -U -qq
    !pip install fastai -U -qq
    !pip install wandb -U -qq

else:
    # Set base folder path for storing files on local machine
    # REPLACE WITH YOUR FOLDER
    # FILL THIS ONLY IF YOU ARE RUNNING ON A LOCAL MACHINE
    print('Path is /Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Data')

    base_folder = Path('/Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning')
    data_folder = Path('/Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Custom_files')
    !pip install pytorch-lightning==2.0.9 -qq
```

```
    !pip install torchmetrics -U -qq
    !pip install fastdownload -U -qq
    !pip install fastai -U -qq
    !pip install wandb -U -qq
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[2]:
```
# custom_function_folder = base_folder/'data/custom-functions/fall_2023'
# sys.path.append(str(custom_function_folder))
# model_folder = base_folder/'data/models/dl_fall_2023/dog_breed/oct-9'
# model_folder.mkdir(parents=True, exist_ok=True)
# project_folder = base_folder/'data/imagenette2'
# kaggle_api_folder = base_folder/'data/.kaggle'


# Change the custom_function_folder to folder in your Google drive folder
# Make sure you keep the mlp_skip_two_layer.py and shared_utils.py files
from pathlib import Path
import sys

# Determine the storage location based on the execution environment
# If running on Google Colab, use Google Drive as storage
if 'google.colab' in str(get_ipython()):
    custom_function_folder = Path('/content/drive/MyDrive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Custom_files') # Your Google Drive

    sys.path.append(str(custom_function_folder))
    model_folder = Path('/content/drive/MyDrive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Data') # Google drive folder where you want␣
 ↪to save model and logs
    model_folder.mkdir(parents=True, exist_ok=True)
    project_folder = Path('/content/drive/MyDrive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Class/Class - 6/Imagenette_project')
    kaggle_api_folder = base_folder/'data/.kaggle'

# If running locally, specify a different path
else:
    # Set base folder path for storing files on local machine
    # REPLACE WITH YOUR FOLDER
    # FILL THIS ONLY IF YOU ARE RUNNING ON A LOCAL MACHINE
    print('Path is /Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Custom_files')
    custom_function_folder = Path('/Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Custom_files') # Your Google Drive
```

```
    sys.path.append(str(custom_function_folder))
    model_folder = Path('/Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Data') # Google drive folder where you want␣
 ↪to save model and logs
    model_folder.mkdir(parents=True, exist_ok=True)
    project_folder = Path('/Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Class/Class - 6/Imagenette_project')
    kaggle_api_folder = base_folder/'data/.kaggle'
    # project_folder = Path('/Users/harikrishnadev/Library/CloudStorage/
 ↪GoogleDrive-harikrish0607@gmail.com/My Drive/Colab_Notebooks/
 ↪BUAN_6382_Applied_DeepLearning/Data')
```

[3]:
```
# import Libraries
import yaml

import torch
import torchmetrics
from torchvision import transforms
import pytorch_lightning as pl
from pytorch_lightning import seed_everything
from pytorch_lightning.tuner import Tuner
from pytorch_lightning.callbacks import ModelCheckpoint, EarlyStopping,␣
 ↪LearningRateMonitor
from pytorch_lightning.loggers import CSVLogger, WandbLogger
import wandb
import gc

from data_module_imagenette2 import ImagenetteDataModule
from multiclass_lightning_module_v0 import MultiClassLightningModule
from model_two_layer_bn import TwoLayerMLPBN
from shared_utils import  plot_losses_acc
```

[4]:
```
plot_losses_acc??
```

2 Function to load the model

[5]:
```
# Function to load the model
def load_model(config):
    model = TwoLayerMLPBN(**config)
    return model
```

3 Functions for Transformations

[6]:
```
def get_train_transforms(resize_height, resize_width, normalize_mean,␣
↪normalize_std):

    return transforms.Compose(
        [
            transforms.Resize((resize_height, resize_width)),
            transforms.ToTensor(),
            transforms.Normalize(normalize_mean, normalize_std),
        ]
    )

def get_test_transforms(resize_height, resize_width, normalize_mean,␣
↪normalize_std):

    return transforms.Compose(
        [
            transforms.Resize((resize_height, resize_width)),
            transforms.ToTensor(),
            transforms.Normalize(normalize_mean, normalize_std),
        ]
    )
```

4 Function to load DataModule

[7]:
```
def load_datamodule(config, data_folder):
    # Fetch the correct transform function based on config and pass the␣
↪appropriate arguments
    train_transform = get_train_transforms(**config['train_transform'])
    test_transform = get_test_transforms(**config['test_transform'])
    dm = ImagenetteDataModule(
        data_dir=data_folder,
        train_transform=train_transform,
        test_transform=test_transform,
        **config['data_module']
    )
    return dm
```

5 Function to load LightningModule

[8]:
```
def load_lightning_module(config, model):
    optimizer_cls = eval(config['optimizer_cls'])
    loss_fn = eval(config['loss_fn'])()  # directly instantiate the loss␣
↪function
    metric_cls = eval(config['metric_cls'])
```

```
    # If scheduler is defined, convert its string to class as well
    if config.get('scheduler_cls'):
        scheduler_cls = eval(config['scheduler_cls'])
        scheduler_options = config['scheduler_options']
        scheduler_params =   config['scheduler_params']
    else:
        scheduler_cls = None

    lightning_module = MultiClassLightningModule(model=model,
                                                 optimizer_cls=optimizer_cls,
                                                 loss_fn=loss_fn,
                                                 metric_cls=metric_cls,
                                                 scheduler_cls=scheduler_cls,
                                                ␣
 ↪scheduler_options=scheduler_options,
                                                ␣
 ↪scheduler_params=scheduler_params,
                                                 **config['others']
)
    return lightning_module
```

6 Function to load the Trainer

[9]:
```
def load_trainer(model, trainer_config, cl_config, batch_size, model_folder, ␣
 ↪logging=False, checkpointing=True, early_stopping=False):

    lr_monitor = LearningRateMonitor(**cl_config['lr_monitor'])
    callbacks = [lr_monitor]
    if checkpointing:
        model_checkpoint_callback = ModelCheckpoint(dirpath=model_folder/
 ↪cl_config['log_dir'],
                                                    **cl_config['model_checkpoint'])
        callbacks.append(model_checkpoint_callback)

    if early_stopping:
        early_stop_callback = EarlyStopping(**cl_config['early_stopping'] )
        callbacks.append(early_stop_callback)

    if logging:
        # For WandB logger:
        wandb_logger = WandbLogger(project=cl_config['wandb']['project'],␣
 ↪name=cl_config['wandb']['name'], save_dir=model_folder/cl_config['log_dir'])
        wandb_logger.experiment.config.update({'batch_size': batch_size,␣
 ↪'epochs': trainer_config['max_epochs']})
        wandb_logger.watch(model)
```

```
        # For CSV logger:
        csv_logger = CSVLogger(save_dir=model_folder/cl_config['log_dir'],␣
 ↪name=cl_config['csv']['name'])
        csv_logger.log_hyperparams(params={'batch_size': batch_size, 'epochs':␣
 ↪trainer_config['max_epochs']})

        trainer = pl.Trainer(callbacks=callbacks,
                             logger=[csv_logger, wandb_logger],
                             **trainer_config)
    else:
        trainer = pl.Trainer(callbacks=callbacks,
                             **trainer_config
                )
    return trainer
```

7 Function to load components

[10]:
```
def load_components(model_config, data_module_config, lightning_module_config,␣
 ↪data_folder, trainer_config,
cl_config, batch_size,logging=False, checkpointing=True, early_stopping=False):

    # Load the model
    model = load_model(model_config)

    # Load the data module
    dm = load_datamodule(data_module_config, data_folder)

    # Load the lightning module
    lightning_module = load_lightning_module(lightning_module_config, model)

    # Load the trainer
    trainer = load_trainer(model, trainer_config, cl_config, batch_size,␣
 ↪model_folder,  logging=logging,
                           checkpointing=checkpointing,␣
 ↪early_stopping=early_stopping)

    return model, dm, lightning_module, trainer
```

[11]:
```
def load_yaml(filepath):
    with open(filepath, 'r') as file:
        return yaml.safe_load(file)
```

8 Function to Load config files

[12]:
```
# Load configurations from YAML files
def load_all_configs():
    model_config = load_yaml(project_folder/'model_config.yaml')
    data_module_config = load_yaml(project_folder/'data_module_config.yaml')
    lightning_module_config = load_yaml(project_folder/'lightning_module_config.
 ↪yaml')
    cl_config = load_yaml(project_folder/'callbacks_loggers_config.yaml')
    trainer_config = load_yaml(project_folder/'trainer_config.yaml')

    return model_config, data_module_config, lightning_module_config,␣
 ↪cl_config, trainer_config
```

9 Function to free memory

[13]:
```
def free_memory():
    """
    Attempts to free up memory by deleting variables and running Python's␣
 ↪garbage collector.
    """
    gc.collect()
    for device_id in range(torch.cuda.device_count()):
        torch.cuda.set_device(device_id)
        torch.cuda.empty_cache()
    gc.collect()
```

10 Run One training and validation batch to check bugs

[14]:
```
# Load components
free_memory()
seed_everything(42)
model_config, data_module_config, lightning_module_config, cl_config,␣
 ↪trainer_config = load_all_configs()
# override default values
trainer_config['fast_dev_run']=True
model, dm, lightning_module, trainer = load_components(model_config,␣
 ↪data_module_config,
                                                       lightning_module_config,␣
 ↪data_folder, trainer_config,
                                                        cl_config,␣
 ↪batch_size=data_module_config['data_module']['batch_size'],
                                                       logging=False,␣
 ↪checkpointing=False, early_stopping=False)
dm.prepare_data()
```

```
trainer.fit(lightning_module, dm)
```

```
INFO:lightning_fabric.utilities.seed:Global seed set to 42
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used:
True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU
cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:Running in `fast_dev_run` mode: will
run the requested loop using 1 batch(es). Logging and checkpointing is
suppressed.
```

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

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

```
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type               | Params
-------------------------------------------------
0 | model        | TwoLayerMLPBN      | 281 M
1 | loss_fn      | CrossEntropyLoss   | 0
2 | train_metric | MulticlassAccuracy | 0
3 | val_metric   | MulticlassAccuracy | 0
4 | test_metric  | MulticlassAccuracy | 0
-------------------------------------------------
281 M     Trainable params
0         Non-trainable params
281 M     Total params
1,126.032 Total estimated model params size (MB)
```

```
Training: 0it [00:00, ?it/s]
```

```
Validation: 0it [00:00, ?it/s]
```

```
Epoch 1: Val_Loss: 2.97, Val_Metric: 0.20 |
```

```
INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_steps=1`
reached.
```

```
Train_Loss: 2.37, Train_Metric: 0.05
```

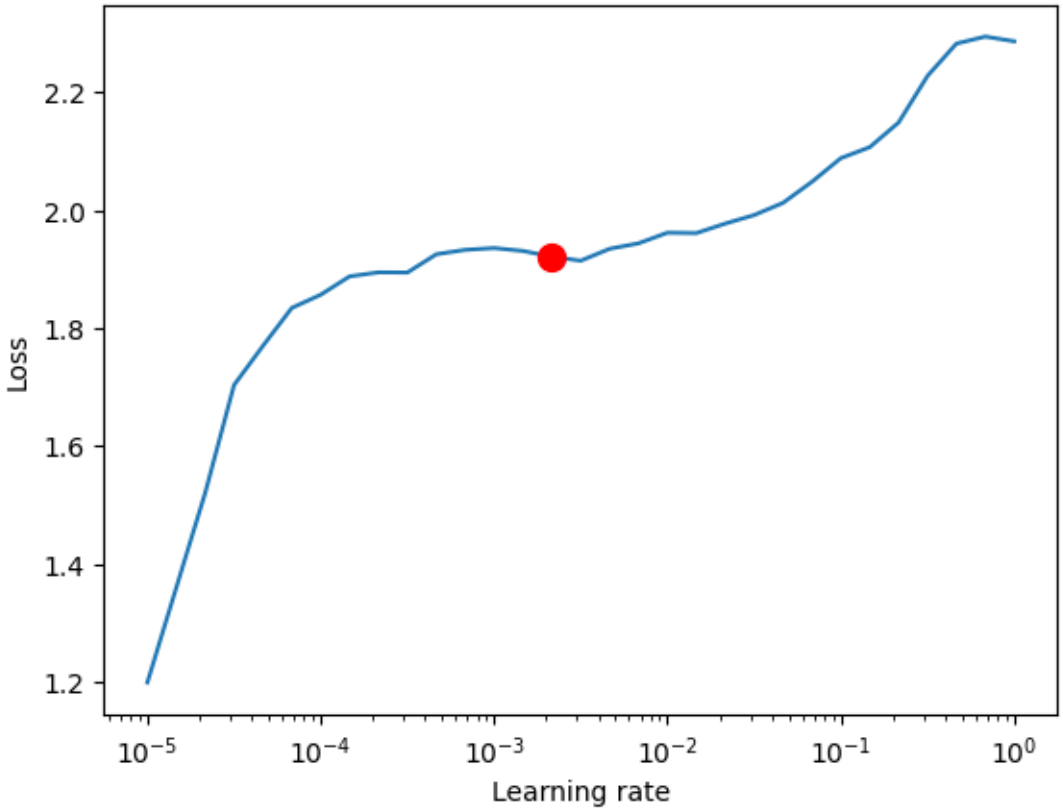
11 Find Learning Rate

[15]:
```
# Load components
free_memory()
seed_everything(42)
model_config, data_module_config, lightning_module_config, cl_config,␣
  ↪trainer_config = load_all_configs()
```

```
# override default values
trainer_config['max_epochs']=3
data_module_config['data_module']['batch_size']=64

model, dm, lightning_module, trainer = load_components(model_config,␣
 ↪data_module_config,
                                                        lightning_module_config,␣
 ↪data_folder, trainer_config,
                                                         cl_config,␣
 ↪batch_size=data_module_config['data_module']['batch_size'],
                                                         logging=False,␣
 ↪checkpointing=False, early_stopping=False)
dm.setup()
tuner = Tuner(trainer)
lr_finder = tuner.lr_find(lightning_module, datamodule=dm, min_lr=1e-5,␣
 ↪max_lr=1, num_training=30, mode='exponential')
fig = lr_finder.plot(suggest=True)
new_lr = lr_finder.suggestion()
print(new_lr)
```

```
INFO:lightning_fabric.utilities.seed:Global seed set to 42
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used:
True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU
cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_train_batches=1.0)`
was configured so 100% of the batches per epoch will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_val_batches=1.0)` was
configured so 100% of the batches will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_test_batches=1.0)` was
configured so 100% of the batches will be used..
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]

Epoch 1: Val_Loss: 2.30, Val_Metric: 0.11 |

Finding best initial lr:   0%|          | 0/30 [00:00<?, ?it/s]

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_steps=30`
reached.
INFO:pytorch_lightning.tuner.lr_finder:Learning rate set to 0.002154434690031884
INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/.lr_find_eaf3f73e-1030-44fa-9f5f-7df4a580252a.ckpt

Train_Loss: 2.29, Train_Metric: 0.27

INFO:pytorch_lightning.utilities.rank_zero:Restored all states from the
checkpoint at /content/.lr_find_eaf3f73e-1030-44fa-9f5f-7df4a580252a.ckpt
```

0.002154434690031884



12 Overfit Small Subset

[16]:
```
# Load components

free_memory()
seed_everything(42)
model_config, data_module_config, lightning_module_config, cl_config,␣
 ↪trainer_config = load_all_configs()

# override default values
data_module_config['data_module']['batch_size']=128
trainer_config['overfit_batches']=1
lightning_module_config['others']['learning_rate']=0.002
trainer_config['max_epochs']=3
model, dm, lightning_module, trainer = load_components(model_config,␣
 ↪data_module_config,
                                                       lightning_module_config,␣
 ↪data_folder, trainer_config,
```

```
                                                        cl_config,␣
 ↪batch_size=data_module_config['data_module']['batch_size'],
                                                        logging=False,␣
 ↪checkpointing=False, early_stopping=False)
dm.setup()
trainer.fit(lightning_module, dm)
```

```
INFO:lightning_fabric.utilities.seed:Global seed set to 42
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used:
True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU
cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(overfit_batches=1)` was
configured so 1 batch will be used.
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_test_batches=1.0)` was
configured so 100% of the batches will be used..
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type               | Params
-------------------------------------------------
0 | model        | TwoLayerMLPBN      | 281 M
1 | loss_fn      | CrossEntropyLoss   | 0
2 | train_metric | MulticlassAccuracy | 0
3 | val_metric   | MulticlassAccuracy | 0
4 | test_metric  | MulticlassAccuracy | 0
-------------------------------------------------
281 M     Trainable params
0         Non-trainable params
281 M     Total params
1,126.032 Total estimated model params size (MB)

Sanity Checking: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 2.30, Val_Metric: 0.13 |

/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/trainer/connectors/data_connector.py:262:
UserWarning: You requested to overfit but enabled train dataloader shuffling. We
are turning off the train dataloader shuffling for you.
  rank_zero_warn(

Training: 0it [00:00, ?it/s]

Validation: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 63.51, Val_Metric: 0.14 | Train_Loss: 2.43, Train_Metric:
0.07
```

```
Validation: 0it [00:00, ?it/s]

Epoch 2: Val_Loss: 4.32, Val_Metric: 0.09 | Train_Loss: 1.81, Train_Metric: 0.55

Validation: 0it [00:00, ?it/s]

Epoch 3: Val_Loss: 3.19, Val_Metric: 0.10 | Train_Loss: 1.17, Train_Metric: 0.86

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=3`
reached.
```

13 Regularization -I

- Early stopping with a patience of 5, total epochs = 50
- Gradient Clipping
- Weight Deacay of 1
- Reduce Learning rate on plateau
- Use 50% of train/val data

[17]:
```
free_memory()
seed_everything(42)
model_config, data_module_config, lightning_module_config, cl_config,␣
 ↪trainer_config = load_all_configs()

# override default values
data_module_config['data_module']['batch_size']=128
lightning_module_config['others']['learning_rate']=0.002
trainer_config['max_epochs']=3
trainer_config['gradient_clip_val']=2
trainer_config['log_every_n_steps']=20

lightning_module_config['others']['optimizer_params']['weight_decay']=1
lightning_module_config['others']['learning_rate']=0.002
lightning_module_config['scheduler_cls']='torch.optim.lr_scheduler.
 ↪ReduceLROnPlateau'
lightning_module_config['scheduler_params']= {'mode': 'max', 'patience': 0,␣
 ↪'factor': 0.5, 'verbose': True}
lightning_module_config['scheduler_options']= {'monitor': 'val_loss',␣
 ↪'interval': 'epoch', 'frequency': 1}
cl_config['lr_monitor']['logging_interval']='epoch'
cl_config['wandb']['project']='imagenette2_multiclass'
cl_config['wandb']['name']='two_layer_mlp_bn_v0'

data_module_config['data_module']['small_subset']=True
data_module_config['data_module']['num_samples_small']=0.5

model, dm, lightning_module, trainer = load_components(model_config,␣
 ↪data_module_config,
```

```
                                                        lightning_module_config,␣
 ↪data_folder, trainer_config,
                                                         cl_config,␣
 ↪batch_size=data_module_config['data_module']['batch_size'],
                                                         logging=True,␣
 ↪checkpointing=True, early_stopping=True)
dm.setup()
trainer.fit(lightning_module, dm)
```

```
INFO:lightning_fabric.utilities.seed:Global seed set to 42
```

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

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

```
VBox(children=(Label(value='Waiting for wandb.init()…\r'), FloatProgress(value=0.
 ↪011112350955555586, max=1.0…
```

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

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

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

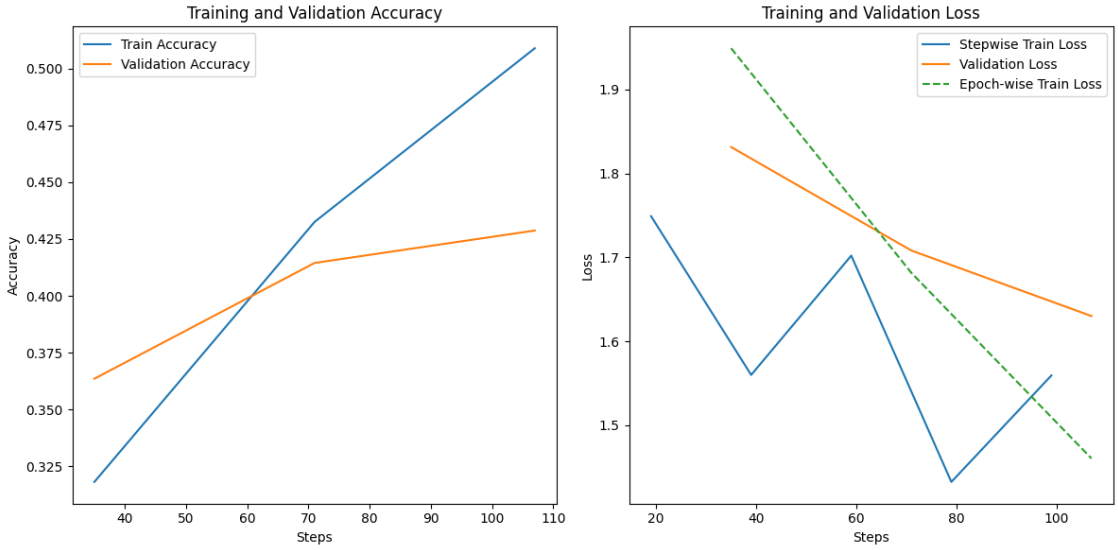
```
<IPython.core.display.HTML object>
```

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

```
wandb: logging graph, to disable use `wandb.watch(log_graph=False)`
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used:
True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU
cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_train_batches=1.0)`
was configured so 100% of the batches per epoch will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_val_batches=1.0)` was
configured so 100% of the batches will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_test_batches=1.0)` was
configured so 100% of the batches will be used..
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/callbacks/model_checkpoint.py:617: UserWarning:
Checkpoint directory
/content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Data/logs
exists and is not empty.
  rank_zero_warn(f"Checkpoint directory {dirpath} exists and is not empty.")
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type               | Params
```

```
-----------------------------------------------------
0 | model        | TwoLayerMLPBN      | 281 M
1 | loss_fn      | CrossEntropyLoss   | 0
2 | train_metric | MulticlassAccuracy | 0
3 | val_metric   | MulticlassAccuracy | 0
4 | test_metric  | MulticlassAccuracy | 0
-----------------------------------------------------
281 M     Trainable params
0         Non-trainable params
281 M     Total params
1,126.032 Total estimated model params size (MB)

Sanity Checking: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 2.31, Val_Metric: 0.10 |

Training: 0it [00:00, ?it/s]

Validation: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 1.83, Val_Metric: 0.36 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved. New
best score: 0.364

Train_Loss: 1.95, Train_Metric: 0.32

Validation: 0it [00:00, ?it/s]

Epoch 2: Val_Loss: 1.71, Val_Metric: 0.41 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved by
0.051 >= min_delta = 0.0. New best score: 0.414

Train_Loss: 1.68, Train_Metric: 0.43
Epoch 00002: reducing learning rate of group 0 to 1.0000e-03.

Validation: 0it [00:00, ?it/s]

Epoch 3: Val_Loss: 1.63, Val_Metric: 0.43 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved by
0.014 >= min_delta = 0.0. New best score: 0.429

Train_Loss: 1.46, Train_Metric: 0.51

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=3`
reached.

Epoch 00003: reducing learning rate of group 0 to 5.0000e-04.
```

[18]:
```
file = f"{trainer.logger.log_dir}/metrics.csv"
# !ls -la {str(file)}
plot_losses_acc(file)
```



[19]:
```
ckpt_path = trainer.checkpoint_callback.best_model_path
train_acc = trainer.validate(dataloaders=dm.train_dataloader(),␣
 ↪ckpt_path=ckpt_path, verbose=False)
valid_acc = trainer.validate(dataloaders=dm.val_dataloader(),␣
 ↪ckpt_path=ckpt_path, verbose=False)
print(f"Train Accuracy: {train_acc[0]['val_metric']*100:0.2f}")
print(f"Validation Accuracy: {valid_acc[0]['val_metric']*100:0.2f}")
wandb.finish()
```

```
INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=108-v1.ckpt
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=108-v1.ckpt
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/trainer/connectors/data_connector.py:490:
PossibleUserWarning: Your `val_dataloader`'s sampler has shuffling enabled, it
is strongly recommended that you turn shuffling off for val/test dataloaders.
  rank_zero_warn(

Validation: 0it [00:00, ?it/s]

Epoch 4: Val_Loss: 1.31, Val_Metric: 0.57 |

INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=108-v1.ckpt
```

```
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=108-v1.ckpt

Validation: 0it [00:00, ?it/s]

Epoch 4: Val_Loss: 1.63, Val_Metric: 0.43 | Train Accuracy: 56.71
Validation Accuracy: 42.87

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

14 Regularization -II

- Increase the weight decay to 10

[20]:
```
lightning_module_config
```

[20]:
```
{'optimizer_cls': 'torch.optim.AdamW',
 'loss_fn': 'torch.nn.CrossEntropyLoss',
 'metric_cls': 'torchmetrics.Accuracy',
 'scheduler_cls': 'torch.optim.lr_scheduler.ReduceLROnPlateau',
 'scheduler_options': {'monitor': 'val_loss',
  'interval': 'epoch',
  'frequency': 1},
 'scheduler_params': {'mode': 'max',
  'patience': 0,
  'factor': 0.5,
  'verbose': True},
 'others': {'optimizer_params': {'weight_decay': 1},
  'num_classes': 10,
  'learning_rate': 0.002,
  'log_every_n_steps': 1,
  'log_test_metrics': True,
  'display_metrics': True}}
```

[21]:
```
lightning_module_config['others']['optimizer_params']['weight_decay']=10
```

[22]:
```
lightning_module_config
```

[22]:
```
{'optimizer_cls': 'torch.optim.AdamW',
 'loss_fn': 'torch.nn.CrossEntropyLoss',
 'metric_cls': 'torchmetrics.Accuracy',
```

```
 'scheduler_cls': 'torch.optim.lr_scheduler.ReduceLROnPlateau',
 'scheduler_options': {'monitor': 'val_loss',
  'interval': 'epoch',
  'frequency': 1},
 'scheduler_params': {'mode': 'max',
  'patience': 0,
  'factor': 0.5,
  'verbose': True},
 'others': {'optimizer_params': {'weight_decay': 10},
  'num_classes': 10,
  'learning_rate': 0.002,
  'log_every_n_steps': 1,
  'log_test_metrics': True,
  'display_metrics': True}}
```

[23]:
```
# Regularization -II
free_memory()
seed_everything(42)

model, dm, lightning_module, trainer = load_components(model_config,␣
 ↪data_module_config,
                                                        lightning_module_config,␣
 ↪data_folder, trainer_config,
                                                         cl_config,␣
 ↪batch_size=data_module_config['data_module']['batch_size'],
                                                         logging=True,␣
 ↪checkpointing=True, early_stopping=True)
dm.setup()
trainer.fit(lightning_module, dm)
file = f"{trainer.logger.log_dir}/metrics.csv"
print(file)
# plot_losses_acc(file)
ckpt_path = trainer.checkpoint_callback.best_model_path
train_acc = trainer.validate(dataloaders=dm.train_dataloader(),␣
 ↪ckpt_path=ckpt_path, verbose=False)
valid_acc = trainer.validate(dataloaders=dm.val_dataloader(),␣
 ↪ckpt_path=ckpt_path, verbose=False)
print(f"Train Accuracy: {train_acc[0]['val_metric']*100:0.2f}")
print(f"Validation Accuracy: {valid_acc[0]['val_metric']*100:0.2f}")
```

```
INFO:lightning_fabric.utilities.seed:Global seed set to 42
wandb: Currently logged in as: harikrish0607
(harikrishnad). Use `wandb login --relogin` to force relogin
```

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

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

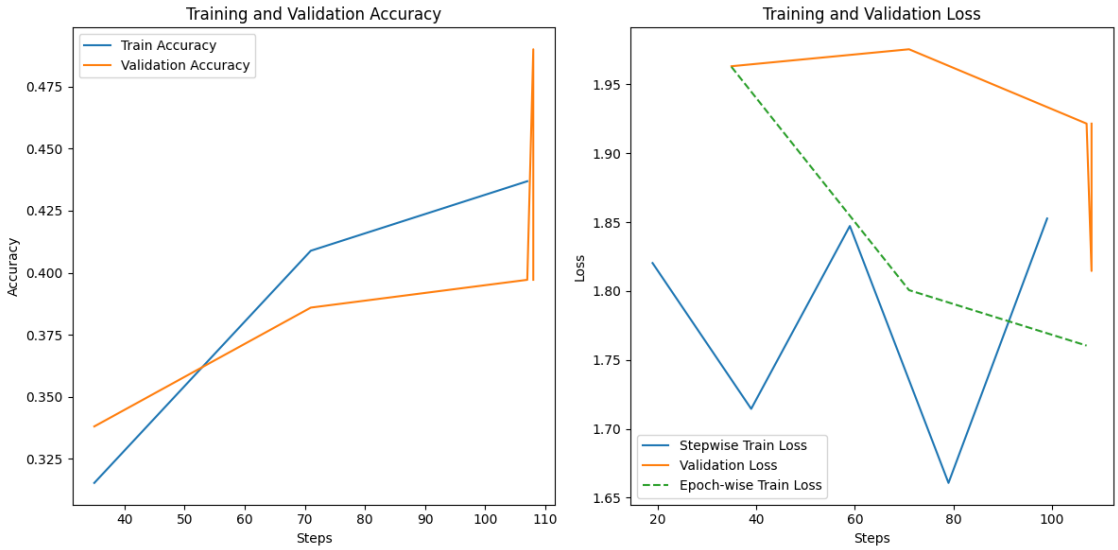
```
<IPython.core.display.HTML object>
```

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

<IPython.core.display.HTML object>

wandb: logging graph, to disable use `wandb.watch(log_graph=False)`
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used:
True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU
cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_train_batches=1.0)`
was configured so 100% of the batches per epoch will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_val_batches=1.0)` was
configured so 100% of the batches will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_test_batches=1.0)` was
configured so 100% of the batches will be used..
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/callbacks/model_checkpoint.py:617: UserWarning:
Checkpoint directory
/content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Data/logs
exists and is not empty.
  rank_zero_warn(f"Checkpoint directory {dirpath} exists and is not empty.")
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type               | Params
-------------------------------------------------
0 | model        | TwoLayerMLPBN      | 281 M
1 | loss_fn      | CrossEntropyLoss   | 0
2 | train_metric | MulticlassAccuracy | 0
3 | val_metric   | MulticlassAccuracy | 0
4 | test_metric  | MulticlassAccuracy | 0
-------------------------------------------------
281 M     Trainable params
0         Non-trainable params
281 M     Total params
1,126.032 Total estimated model params size (MB)

Sanity Checking: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 2.31, Val_Metric: 0.10 |

Training: 0it [00:00, ?it/s]

Validation: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 1.96, Val_Metric: 0.34 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved. New
best score: 0.338
```

```
Train_Loss: 1.96, Train_Metric: 0.32

Validation: 0it [00:00, ?it/s]

Epoch 2: Val_Loss: 1.98, Val_Metric: 0.39 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved by
0.048 >= min_delta = 0.0. New best score: 0.386

Train_Loss: 1.80, Train_Metric: 0.41

Validation: 0it [00:00, ?it/s]

Epoch 3: Val_Loss: 1.92, Val_Metric: 0.40 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved by
0.011 >= min_delta = 0.0. New best score: 0.397

Train_Loss: 1.76, Train_Metric: 0.44

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=3`
reached.

Epoch 00003: reducing learning rate of group 0 to 1.0000e-03.

INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=108-v2.ckpt

/content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Data/logs/
csvlogger/version_28/metrics.csv

INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=108-v2.ckpt
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/trainer/connectors/data_connector.py:490:
PossibleUserWarning: Your `val_dataloader`'s sampler has shuffling enabled, it
is strongly recommended that you turn shuffling off for val/test dataloaders.
  rank_zero_warn(

Validation: 0it [00:00, ?it/s]

Epoch 4: Val_Loss: 1.81, Val_Metric: 0.49 |

INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=108-v2.ckpt
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=108-v2.ckpt
```

```
Validation: 0it [00:00, ?it/s]

Epoch 4: Val_Loss: 1.92, Val_Metric: 0.40 | Train Accuracy: 49.00
Validation Accuracy: 39.71
```

[25]:
```
plot_losses_acc(file)
```



[24]:
```
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(value=1.0, max…

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

15 HW5 PART A - Complete Regularization -III and IV

16 Regularization -III

- Use one Cycle Learning Rate instead of Reduce Learning Rate on Plateau

[26]:
```
free_memory()
seed_everything(42)
import numpy as np
```

```
model_config, data_module_config, lightning_module_config, cl_config,␣
 ↪trainer_config = load_all_configs()

# override default values
data_module_config['data_module']['batch_size']=128
lightning_module_config['others']['learning_rate']=0.002
trainer_config['gradient_clip_val']=2
trainer_config['log_every_n_steps']=20
trainer_config['max_epochs']=4

lightning_module_config['others']['optimizer_params']['weight_decay']=10


# Setting the scheduler class
lightning_module_config['scheduler_cls'] = 'torch.optim.lr_scheduler.OneCycleLR'

# Parameters for the OneCycleLR
# Note: 'max_lr' is a required parameter for OneCycleLR; you'll need to specify␣
 ↪it based on your needs
lightning_module_config['scheduler_params'] = {'max_lr':1e-3,'steps_per_epoch':
 ↪75, 'final_div_factor': 1e4, 'div_factor': 25.0, 'pct_start':0.3,␣
 ↪'anneal_strategy':'cos','epochs':4}

# Options related to the monitoring of the scheduler (if needed)
lightning_module_config['scheduler_options'] = {'monitor': 'val_loss',␣
 ↪'interval': 'epoch', 'frequency': 1}



model, dm, lightning_module, trainer = load_components(model_config,␣
 ↪data_module_config,
                                                        lightning_module_config,␣
 ↪data_folder, trainer_config,
                                                         cl_config,␣
 ↪batch_size=data_module_config['data_module']['batch_size'],
                                                        logging=True,␣
 ↪checkpointing=True, early_stopping=False) # change here
dm.prepare_data()

trainer.fit(lightning_module, dm)
file = f"{trainer.logger.log_dir}/metrics.csv"
plot_losses_acc(file)
ckpt_path = trainer.checkpoint_callback.best_model_path
train_acc = trainer.validate(dataloaders=dm.train_dataloader(),␣
 ↪ckpt_path=ckpt_path, verbose=False)
```

```
valid_acc = trainer.validate(dataloaders=dm.val_dataloader(),␣
  ↪ckpt_path=ckpt_path, verbose=False)
print(f"Train Accuracy: {train_acc[0]['val_metric']*100:0.2f}")
print(f"Validation Accuracy: {valid_acc[0]['val_metric']*100:0.2f}")
wandb.finish()
```

```
INFO:lightning_fabric.utilities.seed:Global seed set to 42
```

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

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

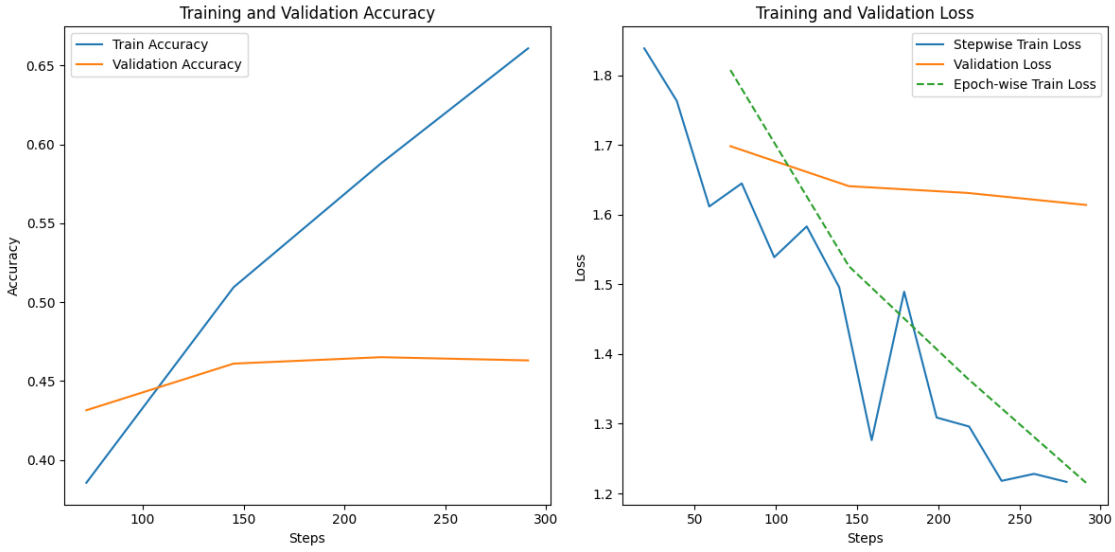
```
<IPython.core.display.HTML object>
```

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

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

```
wandb: logging graph, to disable use `wandb.watch(log_graph=False)`
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used:
True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU
cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_train_batches=1.0)`
was configured so 100% of the batches per epoch will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_val_batches=1.0)` was
configured so 100% of the batches will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_test_batches=1.0)` was
configured so 100% of the batches will be used..
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/callbacks/model_checkpoint.py:617: UserWarning:
Checkpoint directory
/content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Data/logs
exists and is not empty.
  rank_zero_warn(f"Checkpoint directory {dirpath} exists and is not empty.")
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
/usr/local/lib/python3.10/dist-packages/pytorch_lightning/core/optimizer.py:289:
RuntimeWarning: A `OneCycleLR` scheduler is using 'interval': 'epoch'. Are you
sure you didn't mean 'interval': 'step'?
  rank_zero_warn(
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type               | Params
-------------------------------------------------
0 | model        | TwoLayerMLPBN      | 281 M
1 | loss_fn      | CrossEntropyLoss   | 0
2 | train_metric | MulticlassAccuracy | 0
3 | val_metric   | MulticlassAccuracy | 0
4 | test_metric  | MulticlassAccuracy | 0
```

```
----------------------------------------------------
281 M     Trainable params
0         Non-trainable params
281 M     Total params
1,126.032 Total estimated model params size (MB)

Sanity Checking: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 2.31, Val_Metric: 0.09 |

Training: 0it [00:00, ?it/s]

Validation: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 1.70, Val_Metric: 0.43 | Train_Loss: 1.81, Train_Metric: 0.39

Validation: 0it [00:00, ?it/s]

Epoch 2: Val_Loss: 1.64, Val_Metric: 0.46 | Train_Loss: 1.53, Train_Metric: 0.51

Validation: 0it [00:00, ?it/s]

Epoch 3: Val_Loss: 1.63, Val_Metric: 0.47 | Train_Loss: 1.37, Train_Metric: 0.59

Validation: 0it [00:00, ?it/s]

Epoch 4: Val_Loss: 1.61, Val_Metric: 0.46 | Train_Loss: 1.22, Train_Metric: 0.66

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=4`
reached.
```




```
INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=219-v1.ckpt
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
```

```
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=219-v1.ckpt
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/trainer/connectors/data_connector.py:490:
PossibleUserWarning: Your `val_dataloader`'s sampler has shuffling enabled, it
is strongly recommended that you turn shuffling off for val/test dataloaders.
  rank_zero_warn(

Validation: 0it [00:00, ?it/s]

Epoch 5: Val_Loss: 1.24, Val_Metric: 0.66 |

INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=219-v1.ckpt
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=219-v1.ckpt

Validation: 0it [00:00, ?it/s]

Epoch 5: Val_Loss: 1.63, Val_Metric: 0.47 | Train Accuracy: 65.54
Validation Accuracy: 46.51

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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

17 Regularization-1V

- Use one Step LR instead of One Cycler LR

[27]:
```
free_memory()
seed_everything(42)

model_config, data_module_config, lightning_module_config, cl_config,␣
  ↪trainer_config = load_all_configs()

# override default values
data_module_config['data_module']['batch_size']=128
lightning_module_config['others']['learning_rate']=0.002
trainer_config['gradient_clip_val']=2
trainer_config['log_every_n_steps']=20
trainer_config['max_epochs']=3
```

```
lightning_module_config['others']['optimizer_params']['weight_decay']=10


# Setting the scheduler class
lightning_module_config['scheduler_cls'] = 'torch.optim.lr_scheduler.StepLR' #␣
 ↪CODE HERE

# Parameters for the OneCycleLR
# Note: 'max_lr' is a required parameter for OneCycleLR; you'll need to specify␣
 ↪it based on your needs
lightning_module_config['scheduler_params'] = {'step_size':10, 'gamma': 0.5}

# Options related to the monitoring of the scheduler (if needed)
lightning_module_config['scheduler_options'] = {'monitor': 'val_loss',␣
 ↪'interval': 'epoch', 'frequency': 1}



model, dm, lightning_module, trainer = load_components(model_config,␣
 ↪data_module_config,
                                                       lightning_module_config,␣
 ↪data_folder, trainer_config,
                                                        cl_config,␣
 ↪batch_size=data_module_config['data_module']['batch_size'],
                                                       logging=True,␣
 ↪checkpointing=True, early_stopping=True) # change here
dm.prepare_data()

trainer.fit(lightning_module, dm)
file = f"{trainer.logger.log_dir}/metrics.csv"
plot_losses_acc(file)
ckpt_path = trainer.checkpoint_callback.best_model_path
train_acc = trainer.validate(dataloaders=dm.train_dataloader(),␣
 ↪ckpt_path=ckpt_path, verbose=False)
valid_acc = trainer.validate(dataloaders=dm.val_dataloader(),␣
 ↪ckpt_path=ckpt_path, verbose=False)
print(f"Train Accuracy: {train_acc[0]['val_metric']*100:0.2f}")
print(f"Validation Accuracy: {valid_acc[0]['val_metric']*100:0.2f}")
wandb.finish()
```

```
INFO:lightning_fabric.utilities.seed:Global seed set to 42
```

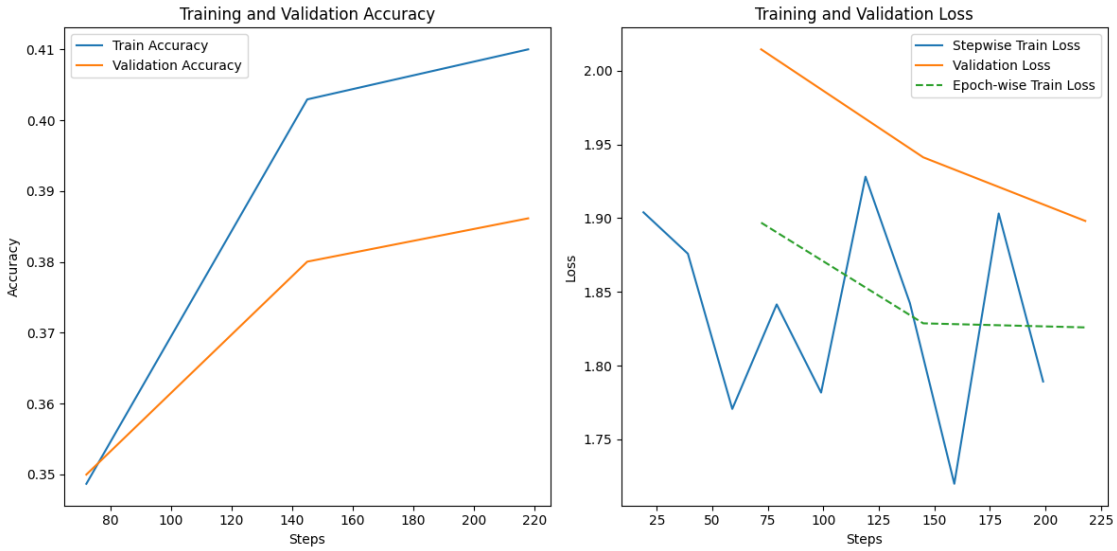
```
<IPython.core.display.HTML object>
```

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

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

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

<IPython.core.display.HTML object>

wandb: logging graph, to disable use `wandb.watch(log_graph=False)`
INFO:pytorch_lightning.utilities.rank_zero:GPU available: True (cuda), used:
True
INFO:pytorch_lightning.utilities.rank_zero:TPU available: False, using: 0 TPU
cores
INFO:pytorch_lightning.utilities.rank_zero:IPU available: False, using: 0 IPUs
INFO:pytorch_lightning.utilities.rank_zero:HPU available: False, using: 0 HPUs
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_train_batches=1.0)`
was configured so 100% of the batches per epoch will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_val_batches=1.0)` was
configured so 100% of the batches will be used..
INFO:pytorch_lightning.utilities.rank_zero:`Trainer(limit_test_batches=1.0)` was
configured so 100% of the batches will be used..
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/callbacks/model_checkpoint.py:617: UserWarning:
Checkpoint directory
/content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Data/logs
exists and is not empty.
  rank_zero_warn(f"Checkpoint directory {dirpath} exists and is not empty.")
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.callbacks.model_summary:
  | Name         | Type               | Params
-------------------------------------------------
0 | model        | TwoLayerMLPBN      | 281 M
1 | loss_fn      | CrossEntropyLoss   | 0
2 | train_metric | MulticlassAccuracy | 0
3 | val_metric   | MulticlassAccuracy | 0
4 | test_metric  | MulticlassAccuracy | 0
-------------------------------------------------
281 M     Trainable params
0         Non-trainable params
281 M     Total params
1,126.032 Total estimated model params size (MB)

Sanity Checking: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 2.31, Val_Metric: 0.09 |

Training: 0it [00:00, ?it/s]

Validation: 0it [00:00, ?it/s]

Epoch 1: Val_Loss: 2.01, Val_Metric: 0.35 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved. New
best score: 0.350
```

```
Train_Loss: 1.90, Train_Metric: 0.35

Validation: 0it [00:00, ?it/s]

Epoch 2: Val_Loss: 1.94, Val_Metric: 0.38 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved by
0.030 >= min_delta = 0.0. New best score: 0.380

Train_Loss: 1.83, Train_Metric: 0.40

Validation: 0it [00:00, ?it/s]

Epoch 3: Val_Loss: 1.90, Val_Metric: 0.39 |

INFO:pytorch_lightning.callbacks.early_stopping:Metric val_metric improved by
0.006 >= min_delta = 0.0. New best score: 0.386

Train_Loss: 1.83, Train_Metric: 0.41

INFO:pytorch_lightning.utilities.rank_zero:`Trainer.fit` stopped: `max_epochs=3`
reached.
```




```
INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=219.ckpt
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=219.ckpt
/usr/local/lib/python3.10/dist-
packages/pytorch_lightning/trainer/connectors/data_connector.py:490:
PossibleUserWarning: Your `val_dataloader`'s sampler has shuffling enabled, it
```

```
is strongly recommended that you turn shuffling off for val/test dataloaders.
  rank_zero_warn(

Validation: 0it [00:00, ?it/s]

Epoch 4: Val_Loss: 1.82, Val_Metric: 0.44 |

INFO:pytorch_lightning.utilities.rank_zero:Restoring states from the checkpoint
path at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearning/Da
ta/logs/epoch=2-step=219.ckpt
INFO:pytorch_lightning.accelerators.cuda:LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES:
[0]
INFO:pytorch_lightning.utilities.rank_zero:Loaded model weights from the
checkpoint at /content/drive/MyDrive/Colab_Notebooks/BUAN_6382_Applied_DeepLearn
ing/Data/logs/epoch=2-step=219.ckpt

Validation: 0it [00:00, ?it/s]

Epoch 4: Val_Loss: 1.90, Val_Metric: 0.39 | Train Accuracy: 43.89
Validation Accuracy: 38.61

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

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