

ml4sci-24-task2-densenet

March 26, 2024

```
[1]: import torch
import numpy as np
import pandas as pd
import pyarrow.parquet as pq
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import timm
import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader, random_split
import torch.nn.functional as F
from torchvision import models

import torch.optim as optim
from tqdm import tqdm

from sklearn.metrics import roc_auc_score, confusion_matrix, roc_curve
import seaborn as sns
```

```
[2]: chunk_size = 15

# List of Parquet file paths
file_paths = [
    '/kaggle/input/task2-24/QCDToGGQQ_IMGjet_RH1all_jet0_run0_n36272.test.
↳snappy.parquet',
    '/kaggle/input/task2-24/QCDToGGQQ_IMGjet_RH1all_jet0_run1_n47540.test.
↳snappy.parquet',
    '/kaggle/input/task2-24/QCDToGGQQ_IMGjet_RH1all_jet0_run2_n55494.test.
↳snappy.parquet'
]

# Initialize an empty list to store dataframes
dfs = []

# Loop through each file path
for file_path in file_paths:
    # Create a Parquet file reader object
```

```

parquet_file = pq.ParquetFile(file_path)

# Determine the total number of rows in the file
total_rows = parquet_file.metadata.num_rows

# Calculate the number of chunks
num_chunks = total_rows // chunk_size + (1 if total_rows % chunk_size else 0)

# Loop over the file in chunks
for chunk_index in range(num_chunks):
    # Read a chunk of rows from the file
    chunk = parquet_file.read_row_group(chunk_index, columns=None)
    df = chunk.to_pandas()

    # Append the DataFrame to the list
    dfs.append(df)

# Concatenate all the DataFrames into a single DataFrame
data = pd.concat(dfs, ignore_index=True)

```

```

[3]: def to_3d(arr):
    vishak=[]
    for i in range (0,3):
        vis=np.stack(np.stack(arr)[i],axis=-1)
        vishak.append(vis)
    vishak=np.array(vishak)
    return vishak

```

```

[4]: data["X_jets"] = data["X_jets"].apply(to_3d)

```

```

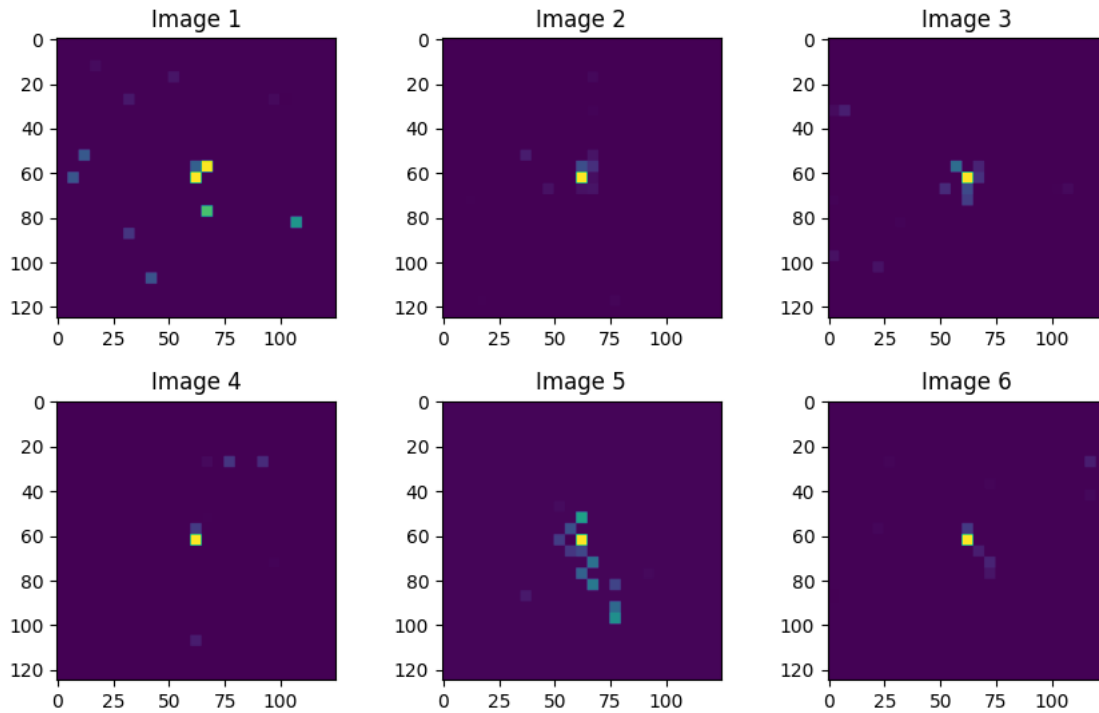
[5]: fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(10, 6))

# Loop over the axes and image ids, and plot each image on a separate subplot
for i, ax in enumerate(axes.flatten()):
    image = data['X_jets'][i][2,:,:]
    ax.imshow(image)
    ax.set_title(f'Image {i+1}')

# Adjust spacing between subplots
plt.subplots_adjust(left=0.1, right=0.9, bottom=0.1, top=0.9, wspace=0.3,
                    hspace=0.3)

# Show the plot
plt.show()

```



```
[6]: data.columns
```

```
[6]: Index(['X_jets', 'pt', 'm0', 'y'], dtype='object')
```

```
[7]: # data['y']
```

```
[8]: class task2Dataset(Dataset):
    def __init__(self, dataframe, transform=None):
        self.dataframe = dataframe
        self.transform = transform

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

    def __getitem__(self, idx):
        # Assuming 'X_jets' column contains paths to images or actual image data
        X = self.dataframe.iloc[idx]['X_jets']
        mean = X.mean(axis=(0, 1, 2), keepdims=True)
        std = X.std(axis=(0, 1, 2), keepdims=True)

        # Normalize each channel separately
        X = (X - mean) / std
        y = self.dataframe.iloc[idx]['y']
```

```

    if self.transform:
        X = self.transform(X)

    # Convert X and y to PyTorch tensors
    X_tensor = torch.tensor(X, dtype=torch.float)
    y_tensor = torch.tensor(y, dtype=torch.long)

    return X_tensor, y_tensor

```

```

[9]: jet_dataset = task2Dataset(dataframe=data)

train_dataset, val_dataset = train_test_split(jet_dataset, test_size=0.2,
↳ random_state=42)

train_loader = DataLoader(dataset=train_dataset, batch_size=256, shuffle=True)
val_loader = DataLoader(dataset=val_dataset, batch_size=32, shuffle=False)

```

```

[10]: next(iter(train_loader))[0].shape

```

```

[10]: torch.Size([256, 3, 125, 125])

```

```

[11]: class CustomDenseNet(nn.Module):
    def __init__(self, num_classes=2, pretrained=True):
        super(CustomDenseNet, self).__init__()

        self.model = timm.create_model('densenet121', pretrained=pretrained,
↳ num_classes=num_classes)

    def forward(self, x):
        return self.model(x)

# Initialize your model
model = CustomDenseNet(num_classes=2, pretrained=True)

# Print your model architecture
print(model)

```

```

model.safetensors: 0%|          | 0.00/32.3M [00:00<?, ?B/s]

```

```

CustomDenseNet(
  (model): DenseNet(
    (features): Sequential(
      (conv0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3),
bias=False)

```

```

(norm0): BatchNormAct2d(
  64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
  (drop): Identity()
  (act): ReLU(inplace=True)
)
(pool0): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1,
ceil_mode=False)
(denseblock1): DenseBlock(
  (denselayer1): DenseLayer(
    (norm1): BatchNormAct2d(
      64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
      (drop): Identity()
      (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
      128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
      (drop): Identity()
      (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
  )
  (denselayer2): DenseLayer(
    (norm1): BatchNormAct2d(
      96, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
      (drop): Identity()
      (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(96, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
      128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
      (drop): Identity()
      (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
  )
  (denselayer3): DenseLayer(
    (norm1): BatchNormAct2d(
      128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
      (drop): Identity()
      (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)

```

```

        (norm2): BatchNormAct2d(
          128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      )
      (denselayer4): DenseLayer(
        (norm1): BatchNormAct2d(
          160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(160, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
          128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      )
      (denselayer5): DenseLayer(
        (norm1): BatchNormAct2d(
          192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(192, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
          128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      )
      (denselayer6): DenseLayer(
        (norm1): BatchNormAct2d(
          224, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(224, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)

```

```

        (norm2): BatchNormAct2d(
          128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      )
    )
    (transition1): DenseTransition(
      (norm): BatchNormAct2d(
        256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
      )
      (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
      (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock2): DenseBlock(
      (denselayer1): DenseLayer(
        (norm1): BatchNormAct2d(
          128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
          128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
      )
      (denselayer2): DenseLayer(
        (norm1): BatchNormAct2d(
          160, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(160, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
          128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
          (drop): Identity()
          (act): ReLU(inplace=True)
        )
      )
    )

```

```

        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer3): DenseLayer(
        (norm1): BatchNormAct2d(
            192, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(192, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer4): DenseLayer(
        (norm1): BatchNormAct2d(
            224, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(224, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): DenseLayer(
        (norm1): BatchNormAct2d(
            256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
    )

```



```

        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): DenseLayer(
        (norm1): BatchNormAct2d(
            288, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(288, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer7): DenseLayer(
        (norm1): BatchNormAct2d(
            320, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(320, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer8): DenseLayer(
        (norm1): BatchNormAct2d(
            352, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(352, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
    )

```

```

        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): DenseLayer(
        (norm1): BatchNormAct2d(
            384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(384, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): DenseLayer(
        (norm1): BatchNormAct2d(
            416, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(416, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer11): DenseLayer(
        (norm1): BatchNormAct2d(
            448, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(448, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    )

```

```

        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): DenseLayer(
        (norm1): BatchNormAct2d(
            480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
        (conv1): Conv2d(480, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    )
    (transition2): DenseTransition(
        (norm): BatchNormAct2d(
            512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock3): DenseBlock(
        (denselayer1): DenseLayer(
            (norm1): BatchNormAct2d(
                256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): DenseLayer(
        (norm1): BatchNormAct2d(

```

```

        288, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(288, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer3): DenseLayer(
        (norm1): BatchNormAct2d(
            320, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(320, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer4): DenseLayer(
        (norm1): BatchNormAct2d(
            352, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(352, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer5): DenseLayer(
        (norm1): BatchNormAct2d(

```

```

        384, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(384, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer6): DenseLayer(
        (norm1): BatchNormAct2d(
            416, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(416, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer7): DenseLayer(
        (norm1): BatchNormAct2d(
            448, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(448, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer8): DenseLayer(
        (norm1): BatchNormAct2d(

```

```

        480, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(480, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer9): DenseLayer(
        (norm1): BatchNormAct2d(
            512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer10): DenseLayer(
        (norm1): BatchNormAct2d(
            544, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(544, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer11): DenseLayer(
        (norm1): BatchNormAct2d(

```

```

        576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(576, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer12): DenseLayer(
        (norm1): BatchNormAct2d(
            608, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(608, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer13): DenseLayer(
        (norm1): BatchNormAct2d(
            640, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(640, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer14): DenseLayer(
        (norm1): BatchNormAct2d(

```

```

        672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(672, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer15): DenseLayer(
    (norm1): BatchNormAct2d(
        704, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(704, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer16): DenseLayer(
    (norm1): BatchNormAct2d(
        736, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(736, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
(denselayer17): DenseLayer(
    (norm1): BatchNormAct2d(

```



```

        768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(768, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer18): DenseLayer(
        (norm1): BatchNormAct2d(
            800, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(800, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer19): DenseLayer(
        (norm1): BatchNormAct2d(
            832, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(832, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer20): DenseLayer(
        (norm1): BatchNormAct2d(

```

```

        864, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(864, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer21): DenseLayer(
        (norm1): BatchNormAct2d(
            896, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(896, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer22): DenseLayer(
        (norm1): BatchNormAct2d(
            928, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(928, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer23): DenseLayer(
        (norm1): BatchNormAct2d(

```

```

        960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(960, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
)
    (denselayer24): DenseLayer(
        (norm1): BatchNormAct2d(
            992, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(992, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
)
    (transition3): DenseTransition(
        (norm): BatchNormAct2d(
            1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (pool): AvgPool2d(kernel_size=2, stride=2, padding=0)
    )
    (denseblock4): DenseBlock(
        (denselayer1): DenseLayer(
            (norm1): BatchNormAct2d(
                512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
                (drop): Identity()
                (act): ReLU(inplace=True)
            )
            (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer2): DenseLayer(
        (norm1): BatchNormAct2d(
            544, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv1): Conv2d(544, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            (drop): Identity()
            (act): ReLU(inplace=True)
        )
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
        )
        (denselayer3): DenseLayer(
            (norm1): BatchNormAct2d(
                576, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
                (drop): Identity()
                (act): ReLU(inplace=True)
            )
            (conv1): Conv2d(576, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
            (norm2): BatchNormAct2d(
                128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
                (drop): Identity()
                (act): ReLU(inplace=True)
            )
            (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
            )
            (denselayer4): DenseLayer(
                (norm1): BatchNormAct2d(
                    608, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
                    (drop): Identity()
                    (act): ReLU(inplace=True)
                )
                (conv1): Conv2d(608, 128, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer5): DenseLayer(
        (norm1): BatchNormAct2d(
            640, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(640, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer6): DenseLayer(
        (norm1): BatchNormAct2d(
            672, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(672, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer7): DenseLayer(
        (norm1): BatchNormAct2d(
            704, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(704, 128, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer8): DenseLayer(
        (norm1): BatchNormAct2d(
            736, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(736, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer9): DenseLayer(
        (norm1): BatchNormAct2d(
            768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(768, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    (denselayer10): DenseLayer(
        (norm1): BatchNormAct2d(
            800, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv1): Conv2d(800, 128, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    (denselayer11): DenseLayer(
        (norm1): BatchNormAct2d(
            832, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
        (conv1): Conv2d(832, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
        (denselayer12): DenseLayer(
            (norm1): BatchNormAct2d(
                864, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            )
            (drop): Identity()
            (act): ReLU(inplace=True)
            (conv1): Conv2d(864, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
            (norm2): BatchNormAct2d(
                128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            )
            (drop): Identity()
            (act): ReLU(inplace=True)
            (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
            (denselayer13): DenseLayer(
                (norm1): BatchNormAct2d(
                    896, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
                )
                (drop): Identity()
                (act): ReLU(inplace=True)
                (conv1): Conv2d(896, 128, kernel_size=(1, 1), stride=(1, 1),

```

```

bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
    )
    (drop): Identity()
    (act): ReLU(inplace=True)
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    (denselayer14): DenseLayer(
        (norm1): BatchNormAct2d(
            928, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
        (conv1): Conv2d(928, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
        (norm2): BatchNormAct2d(
            128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        )
        (drop): Identity()
        (act): ReLU(inplace=True)
        (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
        (denselayer15): DenseLayer(
            (norm1): BatchNormAct2d(
                960, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            )
            (drop): Identity()
            (act): ReLU(inplace=True)
            (conv1): Conv2d(960, 128, kernel_size=(1, 1), stride=(1, 1),
bias=False)
            (norm2): BatchNormAct2d(
                128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
            )
            (drop): Identity()
            (act): ReLU(inplace=True)
            (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
            (denselayer16): DenseLayer(
                (norm1): BatchNormAct2d(
                    992, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
                )
                (drop): Identity()
                (act): ReLU(inplace=True)
                (conv1): Conv2d(992, 128, kernel_size=(1, 1), stride=(1, 1),

```



```

bias=False)
    (norm2): BatchNormAct2d(
        128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    (conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1), bias=False)
    )
    )
    (norm5): BatchNormAct2d(
        1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True
        (drop): Identity()
        (act): ReLU(inplace=True)
    )
    )
    (global_pool): SelectAdaptivePool2d(pool_type=avg,
flatten=Flatten(start_dim=1, end_dim=-1))
    (head_drop): Dropout(p=0.0, inplace=False)
    (classifier): Linear(in_features=1024, out_features=2, bias=True)
    )
    )

```

```

[12]: device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

model.to(device)

criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.0001)
scheduler = optim.lr_scheduler.StepLR(optimizer, step_size=30, gamma=0.1)

```

```

[13]: num_epochs = 20
train_losses, val_losses, val_accuracies = [], [], []
best_loss = 100000
for epoch in range(num_epochs):
    model.train()
    running_loss = 0.0
    train_bar = tqdm(train_loader, desc=f"Epoch {epoch+1}/{num_epochs} [Train]_
↳Loss: 0.0000", leave=False)
    for inputs, labels in train_bar:
        inputs, labels = inputs.to(device), labels.to(device)

        optimizer.zero_grad()

        outputs = model(inputs)
        loss = criterion(outputs, labels)

```

```

        loss.backward()
        optimizer.step()

        running_loss += loss.item() * inputs.size(0)
        train_bar.set_description(f"Epoch {epoch+1}/{num_epochs} [Train] Loss:␣
↪{loss.item():.4f}")

        #scheduler.step()

    epoch_loss = running_loss / len(train_loader.dataset)
    train_losses.append(epoch_loss)

    # Validation phase
    model.eval()
    val_running_loss = 0.0
    correct_predictions = 0
    total_predictions = 0
    val_bar = tqdm(val_loader, desc=f"Epoch {epoch+1}/{num_epochs} [Val] Loss:␣
↪0.0000, Acc: 0.0000", leave=True)

    with torch.no_grad():
        for inputs, labels in val_bar:
            inputs, labels = inputs.to(device), labels.to(device)

            outputs = model(inputs)
            loss = criterion(outputs, labels)
            val_running_loss += loss.item() * inputs.size(0)

            _, predicted = torch.max(outputs, 1)
            correct_predictions += (predicted == labels).sum().item()
            total_predictions += labels.size(0)

        val_bar.set_description(f"Epoch {epoch+1}/{num_epochs} [Val] Loss:␣
↪{loss.item():.4f}, Acc: {correct_predictions/total_predictions:.4f}")

    epoch_val_loss = val_running_loss / len(val_loader.dataset)
    val_losses.append(epoch_val_loss)

    epoch_val_accuracy = correct_predictions / total_predictions
    best_loss = min(epoch_val_loss, best_loss)
    val_accuracies.append(epoch_val_accuracy)

    if(epoch_val_loss== best_loss):

        model_path = f"model_weights_{epoch}.pth"
        torch.save(model.state_dict(), model_path)

```

```
print(f"Epoch {epoch+1}/{num_epochs}, Train Loss: {epoch_loss:.4f}, Val_
↳Loss: {epoch_val_loss:.4f}, Val Accuracy: {epoch_val_accuracy:.4f}")
```

Epoch 1/20 [Val] Loss: 0.1611, Acc: 0.6631: 100%| | 59/59 [00:01<00:00, 32.17it/s]

Epoch 1/20, Train Loss: 0.6360, Val Loss: 0.6440, Val Accuracy: 0.6631

Epoch 2/20 [Val] Loss: 0.1016, Acc: 0.6970: 100%| | 59/59 [00:01<00:00, 37.32it/s]

Epoch 2/20, Train Loss: 0.4891, Val Loss: 0.6004, Val Accuracy: 0.6970

Epoch 3/20 [Val] Loss: 0.0494, Acc: 0.6916: 100%| | 59/59 [00:01<00:00, 37.16it/s]

Epoch 3/20, Train Loss: 0.3794, Val Loss: 0.6139, Val Accuracy: 0.6916

Epoch 4/20 [Val] Loss: 0.0583, Acc: 0.6868: 100%| | 59/59 [00:01<00:00, 36.37it/s]

Epoch 4/20, Train Loss: 0.3047, Val Loss: 0.6420, Val Accuracy: 0.6868

Epoch 5/20 [Val] Loss: 0.0992, Acc: 0.6808: 100%| | 59/59 [00:01<00:00, 37.45it/s]

Epoch 5/20, Train Loss: 0.2390, Val Loss: 0.6990, Val Accuracy: 0.6808

Epoch 6/20 [Val] Loss: 0.0221, Acc: 0.6938: 100%| | 59/59 [00:01<00:00, 37.33it/s]

Epoch 6/20, Train Loss: 0.2097, Val Loss: 0.7190, Val Accuracy: 0.6938

Epoch 7/20 [Val] Loss: 0.0293, Acc: 0.6900: 100%| | 59/59 [00:01<00:00, 37.26it/s]

Epoch 7/20, Train Loss: 0.1547, Val Loss: 0.7540, Val Accuracy: 0.6900

Epoch 8/20 [Val] Loss: 0.0299, Acc: 0.6878: 100%| | 59/59 [00:01<00:00, 37.53it/s]

Epoch 8/20, Train Loss: 0.1586, Val Loss: 0.7887, Val Accuracy: 0.6878

Epoch 9/20 [Val] Loss: 0.0214, Acc: 0.6819: 100%| | 59/59 [00:01<00:00, 37.55it/s]

Epoch 9/20, Train Loss: 0.1036, Val Loss: 0.8087, Val Accuracy: 0.6819

Epoch 10/20 [Val] Loss: 0.0256, Acc: 0.6744: 100%| | 59/59 [00:01<00:00, 37.26it/s]

Epoch 10/20, Train Loss: 0.0687, Val Loss: 0.8698, Val Accuracy: 0.6744

Epoch 11/20 [Val] Loss: 0.0048, Acc: 0.6771: 100%| | 59/59 [00:01<00:00, 36.94it/s]

Epoch 11/20, Train Loss: 0.0750, Val Loss: 0.8997, Val Accuracy: 0.6771

Epoch 12/20 [Val] Loss: 0.0845, Acc: 0.6550: 100%| | 59/59
 [00:01<00:00, 37.54it/s]

Epoch 12/20, Train Loss: 0.1560, Val Loss: 0.8919, Val Accuracy: 0.6550

Epoch 13/20 [Val] Loss: 0.0181, Acc: 0.6706: 100%| | 59/59
 [00:01<00:00, 37.43it/s]

Epoch 13/20, Train Loss: 0.1094, Val Loss: 0.9717, Val Accuracy: 0.6706

Epoch 14/20 [Val] Loss: 0.0615, Acc: 0.6738: 100%| | 59/59
 [00:01<00:00, 37.36it/s]

Epoch 14/20, Train Loss: 0.0545, Val Loss: 1.0072, Val Accuracy: 0.6738

Epoch 15/20 [Val] Loss: 0.0165, Acc: 0.6426: 100%| | 59/59
 [00:01<00:00, 37.52it/s]

Epoch 15/20, Train Loss: 0.2606, Val Loss: 0.9344, Val Accuracy: 0.6426

Epoch 16/20 [Val] Loss: 0.0318, Acc: 0.6555: 100%| | 59/59
 [00:01<00:00, 37.26it/s]

Epoch 16/20, Train Loss: 0.1313, Val Loss: 0.9715, Val Accuracy: 0.6555

Epoch 17/20 [Val] Loss: 0.0094, Acc: 0.6674: 100%| | 59/59
 [00:01<00:00, 37.45it/s]

Epoch 17/20, Train Loss: 0.0736, Val Loss: 1.0079, Val Accuracy: 0.6674

Epoch 18/20 [Val] Loss: 0.0020, Acc: 0.6615: 100%| | 59/59
 [00:01<00:00, 36.83it/s]

Epoch 18/20, Train Loss: 0.1510, Val Loss: 0.9910, Val Accuracy: 0.6615

Epoch 19/20 [Val] Loss: 0.1081, Acc: 0.6690: 100%| | 59/59
 [00:01<00:00, 37.31it/s]

Epoch 19/20, Train Loss: 0.1512, Val Loss: 1.0019, Val Accuracy: 0.6690

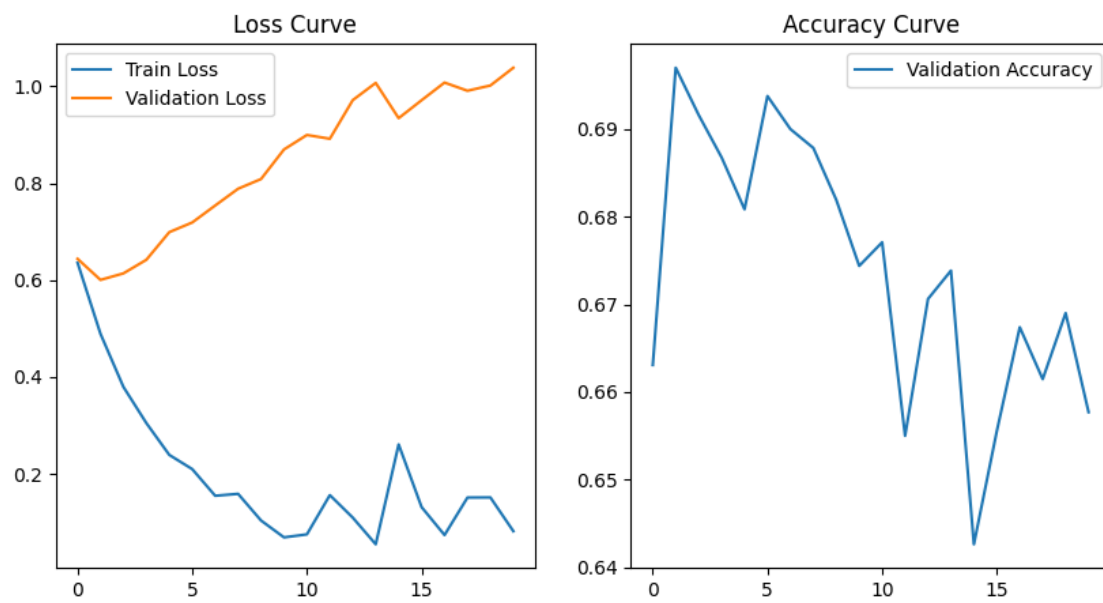
Epoch 20/20 [Val] Loss: 0.0858, Acc: 0.6577: 100%| | 59/59
 [00:01<00:00, 37.38it/s]

Epoch 20/20, Train Loss: 0.0816, Val Loss: 1.0385, Val Accuracy: 0.6577

```
[14]: plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(train_losses, label='Train Loss')
plt.plot(val_losses, label='Validation Loss')
plt.legend()
plt.title('Loss Curve')

plt.subplot(1, 2, 2)
plt.plot(val_accuracies, label='Validation Accuracy')
```

```
plt.legend()
plt.title('Accuracy Curve')
plt.show()
```



[]:

[]: