110066540 陳哲瑋

National Tsing Hua University

Fall 2023 11210IPT 553000

Deep Learning in Biomedical Optical Imaging

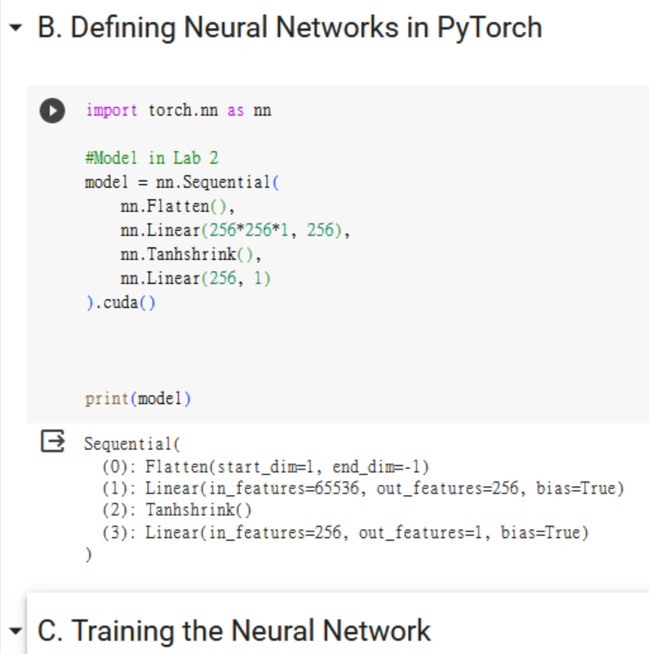
Homework 2

Coding:

1.1 Task A: Transitioning to Cross-Entropy Loss:

要將 BCE 改成 CE 首先需要使用 CrossEntropyLoss 替換掉 BCEWithLogiteLoss，其中 BCE 與 CE 有個最大的差距就是 BCE 是二元分類故要將最後一層 Node 從 1 改為 16。

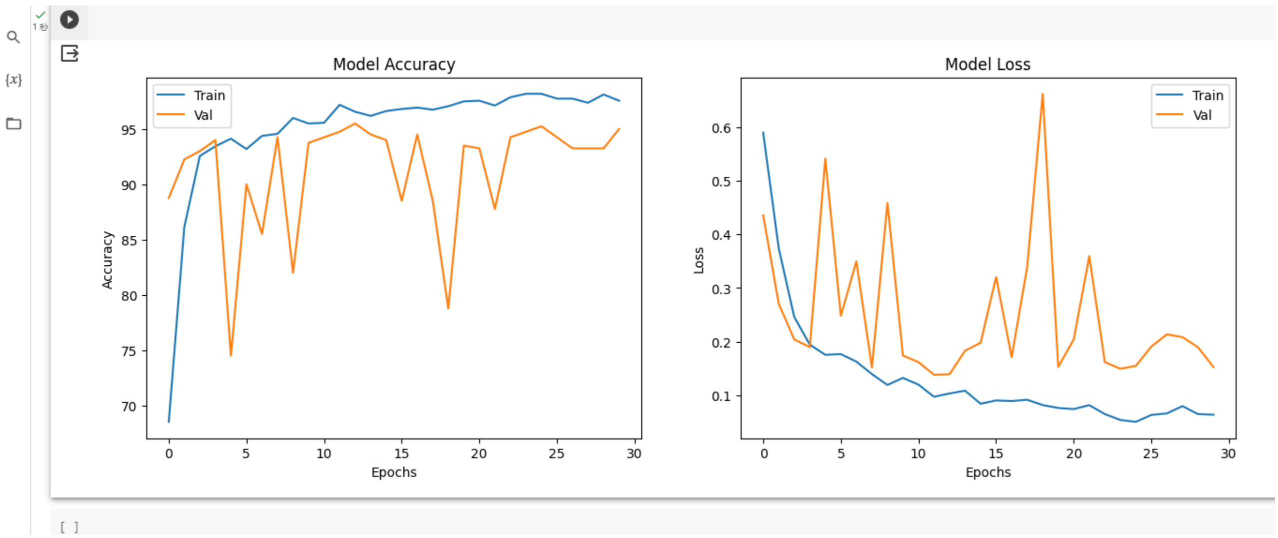

```
a=0
b=0
c=0
d=0
train_losses = []
val_losses = []
train_accuracies = []
val_accuracies = []

epochs = 30
best_val_loss = float('inf')

# Criterion and Optimizer
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=1e-3)
lr_scheduler = CosineAnnealingLR(optimizer, T_max=len(train_loader)*epochs, eta_min=0)
# lr_scheduler = StepLR(optimizer, step_size=10, gamma=0.1)

for epoch in range(epochs):
    # Training
    model.train()
    total_loss = 0.0
```



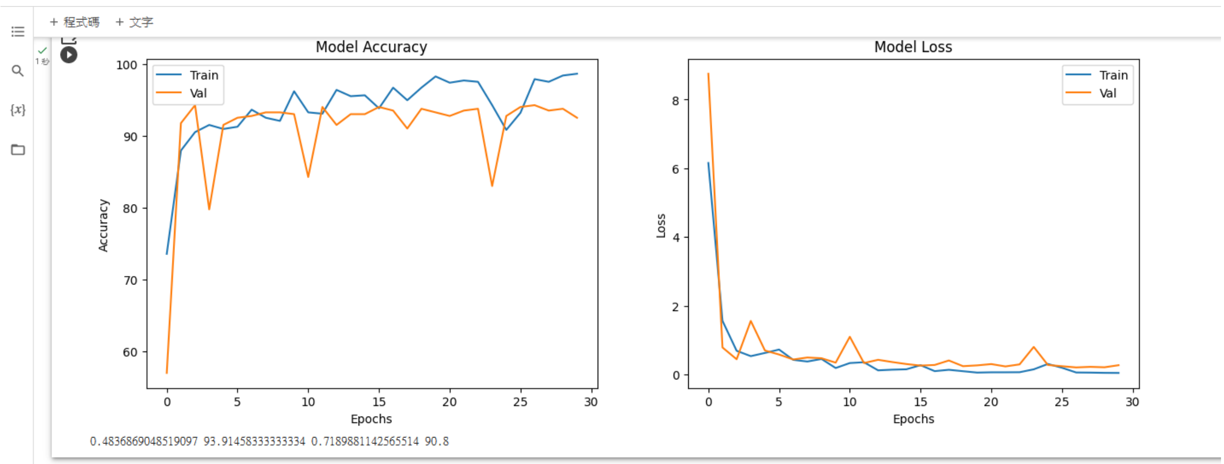

```
import torch.nn as nn

#Model in Lab 2
model = nn.Sequential(
    nn.Flatten(),
    nn.Linear(256*256*1, 256),
    nn.Tanhshrink(),
    nn.Linear(256, 1)
).cuda()



print(model)
```

```
Sequential(
  (0): Flatten(start_dim=1, end_dim=-1)
  (1): Linear(in_features=65536, out_features=256, bias=True)
  (2): Tanhshrink()
  (3): Linear(in_features=256, out_features=1, bias=True)
)
```

1.2 Task B: Creating a Evaluation Code:



上圖為用上課 lab3 的程式跑出來的結果，Train Accuracy 以及 Val Accuracy 之間的差距很大，總合之前的判斷它應該是 over fitting 了，故就用之前的方法將將 neural network 的層數降為一層，下圖為重新訓練的結果，可以看到 Train Accuracy 與 Val Accurac 之間的差距變小，Val Loss 也有下降的跡象。



Report:

2.1 Task A: Performance between BCE loss and BC loss:

在此我們用四項參數的平均值來做分析

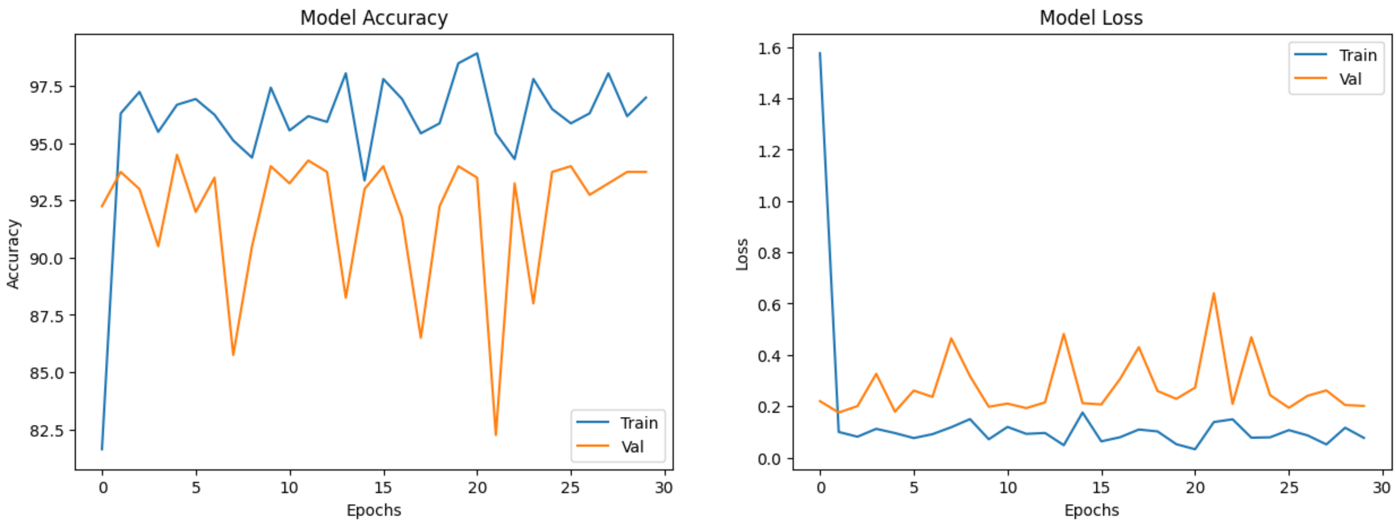
```
print(f'Epoch {epoch+1}/{epochs}, Train Loss: {avg_train_loss:.4f}, Tra
w+=avg_train_loss
x+=train_accuracy
y+=avg_val_loss
z+=val_accuracy
# Learning rate update
lr_scheduler.step()

# Checkpoint
if avg_val_loss < best_val_loss:
    best_val_loss = avg_val_loss
    torch.save(model.state_dict(), 'model_classification.pth')

# Store performance
train_losses.append(avg_train_loss)
train_accuracies.append(train_accuracy)
val_losses.append(avg_val_loss)
val_accuracies.append(val_accuracy)
```

```
Epoch 1/30, Train Loss: 1.5758, Train Accuracy: 81.62%, Val Loss: 0.2198, V
Epoch 2/30, Train Loss: 0.0997, Train Accuracy: 96.31%, Val Loss: 0.1754, V
Epoch 3/30, Train Loss: 0.0811, Train Accuracy: 97.25%, Val Loss: 0.2003, V
Epoch 4/30, Train Loss: 0.1118, Train Accuracy: 95.50%, Val Loss: 0.3263, V
Epoch 5/30, Train Loss: 0.0953, Train Accuracy: 96.69%, Val Loss: 0.1789, V
Epoch 6/30, Train Loss: 0.0759, Train Accuracy: 96.94%, Val Loss: 0.2610, V
Epoch 7/30, Train Loss: 0.0911, Train Accuracy: 96.25%, Val Loss: 0.2364, V
Epoch 8/30, Train Loss: 0.1182, Train Accuracy: 95.12%, Val Loss: 0.4644, V
```

Visualizing model performance

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(1, 2, figsize=(15, 5))

# Plotting training and validation accuracy
ax[0].plot(train_accuracies)
ax[0].plot(val_accuracies)
ax[0].set_title('Model Accuracy')
ax[0].set_xlabel('Epochs')
ax[0].set_ylabel('Accuracy')
ax[0].legend(['Train', 'Val'])

# Plotting training and validation loss
ax[1].plot(train_losses)
ax[1].plot(val_losses)
ax[1].set_title('Model Loss')
ax[1].set_xlabel('Epochs')
ax[1].set_ylabel('Loss')
ax[1].legend(['Train', 'Val'])

plt.show()
print(w/30,x/30,y/30,z/30)
```

下圖為第一次訓練單層 CE 所得到的圖:




```
0.14394941872862785 95.92083333333333 0.2752198615794977 91.96666666666667
```

	avg_train_loss	train_accuracy	avg_val_loss	val_accuracy
BCE	0.1945	97.68%	0.3412	92.46%
CE	0.1439	95.92%	0.2752	91.97%

再經過 4 次訓練後

	avg_train_loss	train_accuracy	avg_val_loss	val_accuracy
BCE	0.0374	97.68%	0.3412	92.46%
CE	0.067	98.22%	0.391	93.23%

BCE 的 avg_train_loss 經過訓練後有所下降，avg_val_loss 、Train Accuracy 以及 Val Accuracy 沒有任何的變化，這代表 BCE 只需要可能 1~2 次訓練就有非常好的效果，就結果來說 CE 的 train_accuracy 及 val_accuracy 都有所上升，證明這個訓練架構是有效的