

## Workshop II 01: Model Training With Early Stop

### Assignment Workshop 01

1. ให้นักศึกษาทำโจทย์ของไฟล์ Workshop II 01.ipynb ให้เรียบร้อยพร้อมทั้งมาตอบคำถามต่อไปนี้

1.1 อธิบายวิธีการเตรียมข้อมูล (Data Preparation) ของนักศึกษาที่ใช้ในการจัดการข้อมูลก่อนที่จะนำเทรนโมเดลครั้งนี้ สามารถแคปภาพหน้าจอแล้วมาอธิบายได้

```
1 !gdown --id 1yCH8zIGo9KjdsRfrvpR1R7mel4uq1GH
2 !unzip '/content/Small_Data_CoV2.zip'
```

```
1 import torch
2 from torch import nn
3 from torch.utils.data import DataLoader
4 from torchvision import datasets
5 from torchvision.transforms import ToTensor, transforms
6 import os
```

- Download ข้อมูลและ Import library ที่จำเป็น

```
1 transform = transforms.Compose([
2     transforms.Resize((224, 224)), # Resize images to 224x224 (common for many models)
3     transforms.ToTensor(), # Convert image to tensor
4     transforms.Normalize(mean=[0.5], std=[0.5]) # Normalize pixel values to [-1, 1]
5 ])
```

- เตรียมการ transform เพื่อใช้สำหรับการแปลงข้อมูลให้อยู่ใน ขนาด 224x224 จากนั้นแปลงเป็น Tensor และ Normalize ข้อมูลให้อยู่ในช่วง -1 ถึง 1 ในการ Normalize กำหนดค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน(SD) ที่ 0.5 เพื่อให้ข้อมูลภาพอยู่ในรูปแบบที่โมเดลเข้าใจได้และช่วยให้การฝึกโมเดลมีประสิทธิภาพมากขึ้น.

```
▶ Small_Data_CoV2_test
▶ Small_Data_CoV2_train
▶ Small_Data_CoV2_val
```

```
7 # Load training, validation, and testing datasets
8 train_data = datasets.ImageFolder(root='Small_Data_CoV2_train', transform=transform)
9 val_data = datasets.ImageFolder(root='Small_Data_CoV2_val', transform=transform)
10 test_data = datasets.ImageFolder(root='Small_Data_CoV2_test', transform=transform)
```

- โหลดชุดข้อมูล (Training, Validation, Testing) โดยใช้ datasets.ImageFolder ในการโหลดข้อมูลที่มีโครงสร้างแบบ Folder กำหนดให้ใช้การ transform ในขั้นตอนก่อนหน้า

```
12 # DataLoaders to iterate through the datasets
13 batch_size = 32
14
15 train_loader = DataLoader(train_data, batch_size=batch_size, shuffle=True)
16 val_loader = DataLoader(val_data, batch_size=batch_size, shuffle=False)
17 test_loader = DataLoader(test_data, batch_size=batch_size, shuffle=False)
```

- ใช้ DataLoader ในการสร้าง Object เพื่อจัดการดึงข้อมูลที่ละ Batch และ Parameter shuffle ใช้ในการกำหนดว่าข้อมูลในแต่ละ dataset จะถูกสุ่มหรือไม่

```
19 # Check if data is loaded correctly (Optional)
20 for images, labels in train_loader:
21     print(f"Batch image shape: {images.shape}")
22     print(f"Batch label shape: {labels.shape}")
23     break
```

- ตรวจสอบข้อมูลจาก train\_loader เพื่อให้แน่ใจว่าแต่ละข้อมูลถูกโหลดอย่างถูกต้อง

```
25 # Classes in your dataset
26 classes = train_data.classes
27 print(f"Classes: {classes}")
```

- ตรวจสอบ Classes ของข้อมูล

1.2 หลักการทำงานของ Early Stopping ในการเทรนโมเดล และมีไว้ทำอะไรได้บ้าง

หลักการของ Early Stopping

- การหยุดการฝึกโมเดลก่อนครบจำนวนรอบ(Epochs)ที่ตั้งไว้ หากโมเดลไม่สามารถปรับปรุงประสิทธิภาพบน Validation Set ได้อีก
- ช่วยป้องกันการ Overfitting โดยลดการฝึกโมเดลเมื่อเริ่มเรียนรู้ข้อมูล Training มากเกินไปจนส่งผลเสียต่อข้อมูล Validation
- ลดเวลาการฝึกโมเดลโดยไม่จำเป็น

- เก็บค่า loss ที่ดีที่สุดไว้ (Loss ที่น้อยสุด)
- นับจำนวนรอบที่ไม่มีการปรับปรุง (epochs\_no\_improve)
- หากค่า Loss ไม่ดีขึ้นเกินจำนวนรอบที่กำหนด (patience) ให้หยุดการฝึกทันที และดึง best\_weight มาใช้

### 1.3 แสดงประสิทธิภาพของโมเดลที่ได้เมื่อไปวัดผลกับข้อมูลทดสอบที่นำมา

```
1 test_loop(test_loader, model, loss_fn)
```

Test Error:

Accuracy: 41.5%, Avg loss: 1.082132

Test Error:

Accuracy: 41.5%, Avg loss: 1.082132