Some common errors in the code

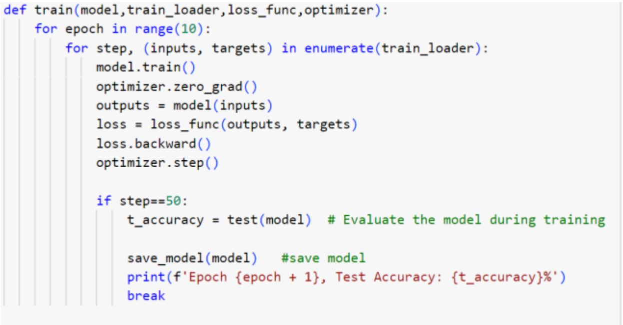
- Need to use same normalization for testing and training

- Forget to implement testing/prediction function

- Forget to switch back to model.train() mode

- Forget to move model to target device
- Forget to move data to target device in testing mode function

- Training epoch not finished. The ‘break’ will jump out off the current training epoch

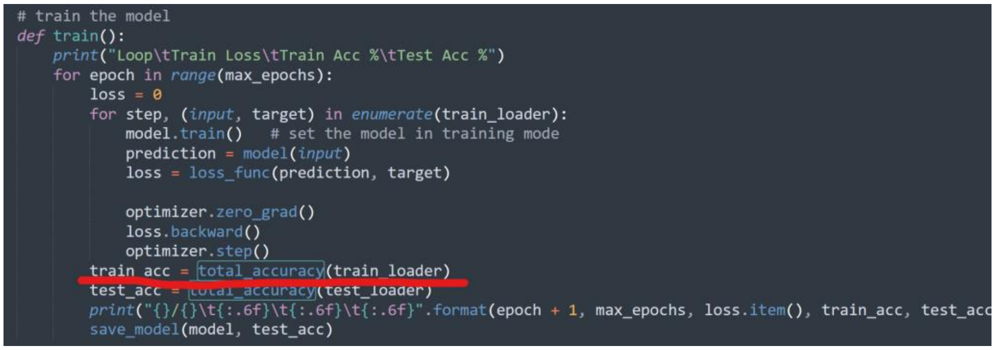


```
def train(model,train_loader,loss_func,optimizer):
    for epoch in range(10):
        for step, (inputs, targets) in enumerate(train_loader):
            model.train()
            optimizer.zero_grad()
            outputs = model(inputs)
            loss = loss_func(outputs, targets)
            loss.backward()
            optimizer.step()

            if step==50:
                t_accuracy = test(model)  # Evaluate the model during training

                save_model(model)   #save model
                print(f'Epoch {epoch + 1}, Test Accuracy: {t_accuracy}%')
                break
```

- To get training accuracy, there is no need to separately go over/inference the training dataset again. You can get the correct or wrong result during each batch of training



```
# train the model
def train():
    print("Loop\tTrain Loss\tTrain Acc %\tTest Acc %")
    for epoch in range(max_epochs):
        loss = 0
        for step, (input, target) in enumerate(train_loader):
            model.train()   # set the model in training mode
            prediction = model(input)
            loss = loss_func(prediction, target)

            optimizer.zero_grad()
            loss.backward()
            optimizer.step()
        train_acc = total_accuracy(train_loader)
        test_acc = total_accuracy(test_loader)
        print("{}/{}\t{:.6f}\t{:.6f}\t{:.6f}".format(epoch + 1, max_epochs, loss.item(), train_acc, test_acc
        save_model(model, test_acc)
```

- Model saved without filename extension. (e.g., ./model/cifar_model) we need .pt extension