AAI3001 Small Project Report

Jurgen Tan

October 19, 2023

Student Information

- Your Name: Jurgen Tan Yu Teng
- Matriculation Number: 2202000

Model Configuration

The following section provides details about the setting used to define the last layer in the model.

Task 1 ResNet50 Multi-Class

```
1
2 class Model(torch.nn.Module):
3   def __init__(self, num_classes, weights=None):
4       super(Model, self).__init__()
5       self.model = resnet50(weights=weights)
6
7       num_features = self.model.fc.in_features
8       self.model.fc = torch.nn.Linear(num_features, num_classes)
9
10  def forward(self, x):
11      x = self.model(x)
12      return x
13
```

Listing 1: Task 1 Model Class

Task 2 ResNet50 Multi-Label

```
1 class MultiLabelModel(torch.nn.Module):
2   def __init__(self, num_classes, weights=None):
3       super(MultiLabelModel, self).__init__()
4       self.model = resnet50(weights=weights)
5
6       num_features = self.model.fc.in_features
7       self.model.fc = torch.nn.Linear(num_features, num_classes)
```

```
8
9   def forward(self, x):
10      x = self.model(x)
11      return torch.sigmoid(x)
```

Listing 2: Task 2 Model Class

Experimental Parameters

The experimental parameters used for training the model are as follows:

- Learning Rate: 0.1, 0.01
- Batch Size: 32
- Seed Values: 0 - using torch.manual_seed(0)
- Num Workers: 4 - Specified in the dataloader
- Epochs: 15 (Task 1) and 10 (Task 2)
- Optimiser Momentum: 0.9

These parameters are necessary to reproduce the training process accurately.

Train/Validation Loss Curves

The following figures show the training and validation loss curves for the specified model setting:

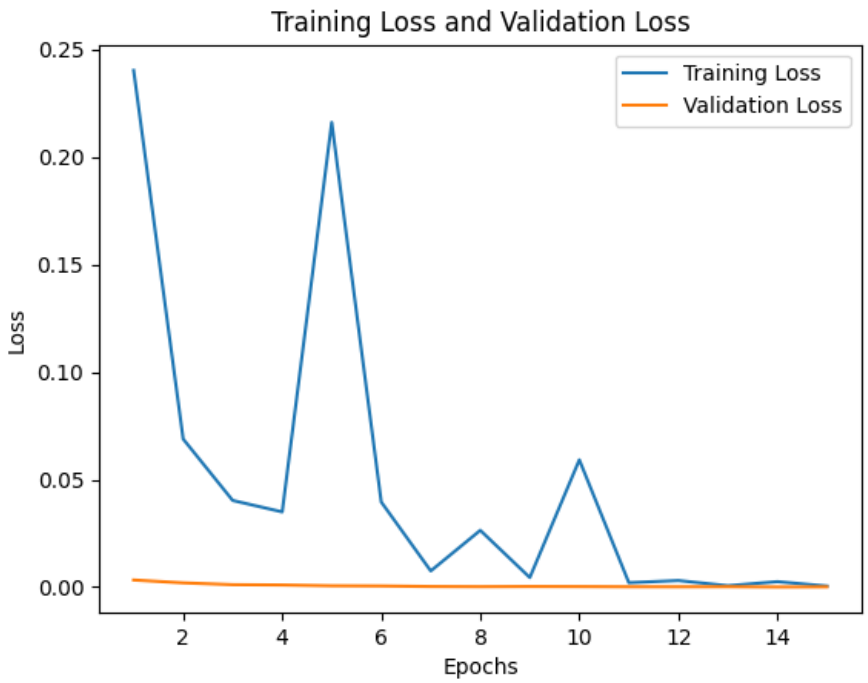

Figure 1: Task 1 (ResNet50 Multi-Class) - Learning Rate 0.01 Augmentation 0

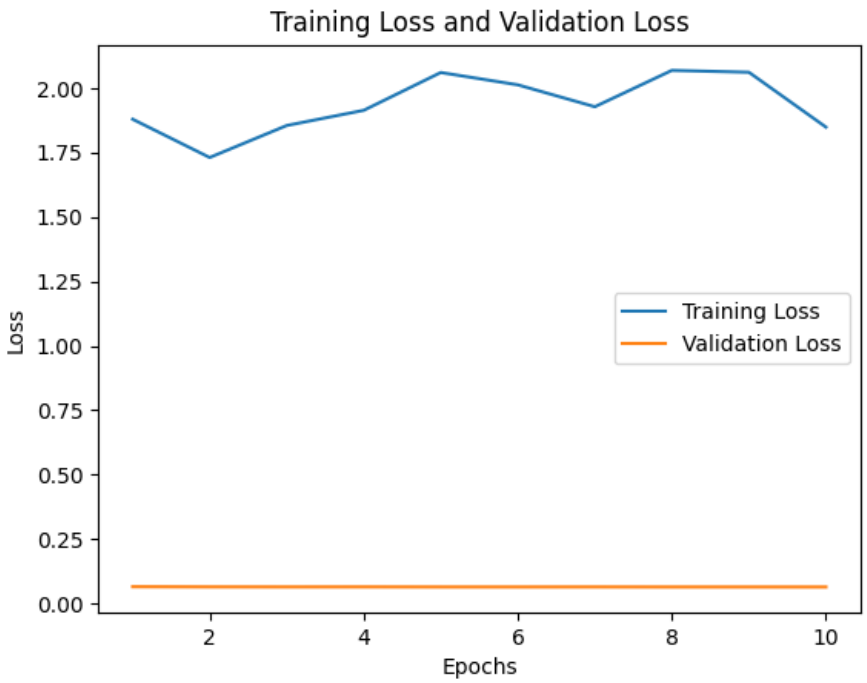

Figure 2: Task 2 (ResNet50 Multi-Label) - Learning Rate 0.01