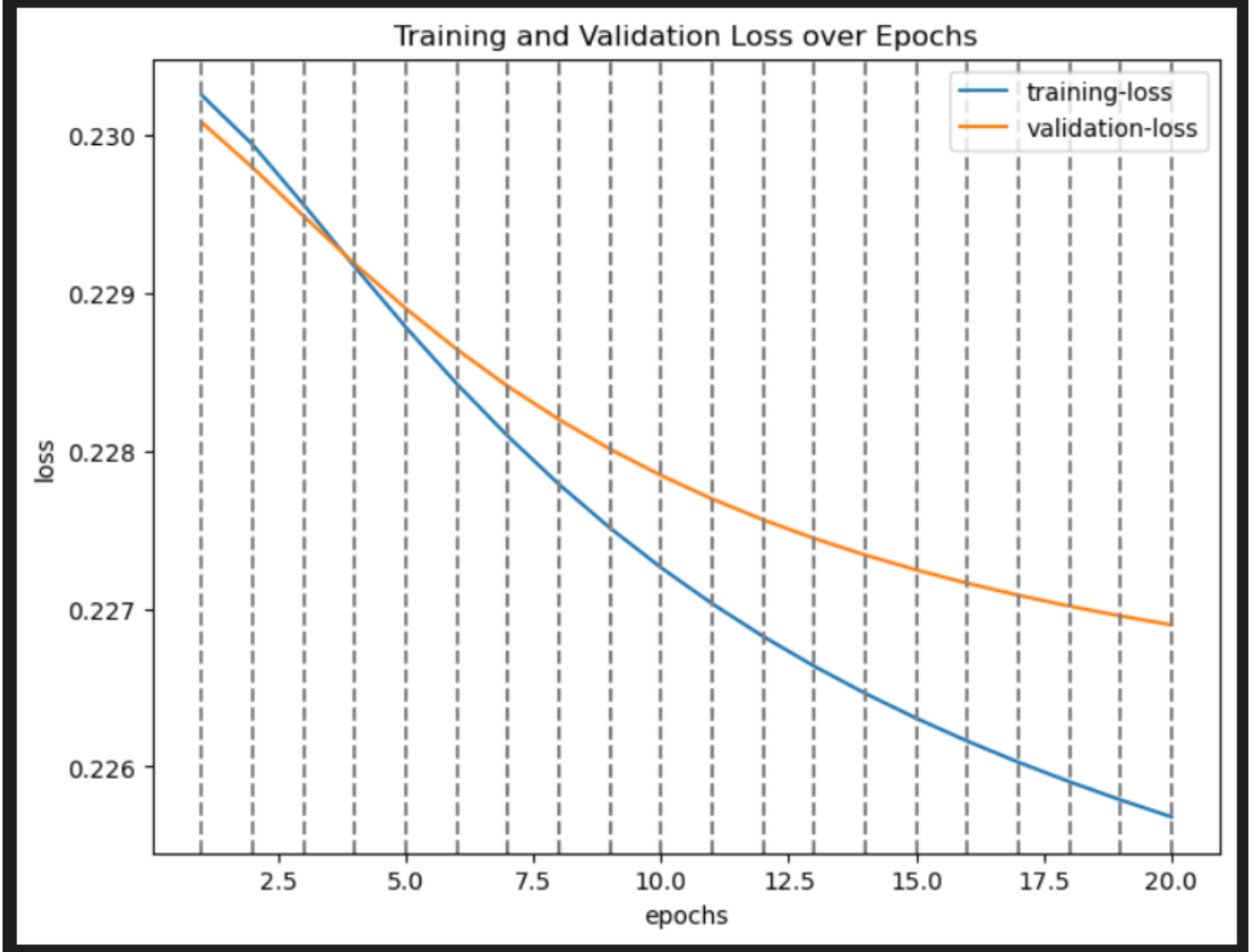
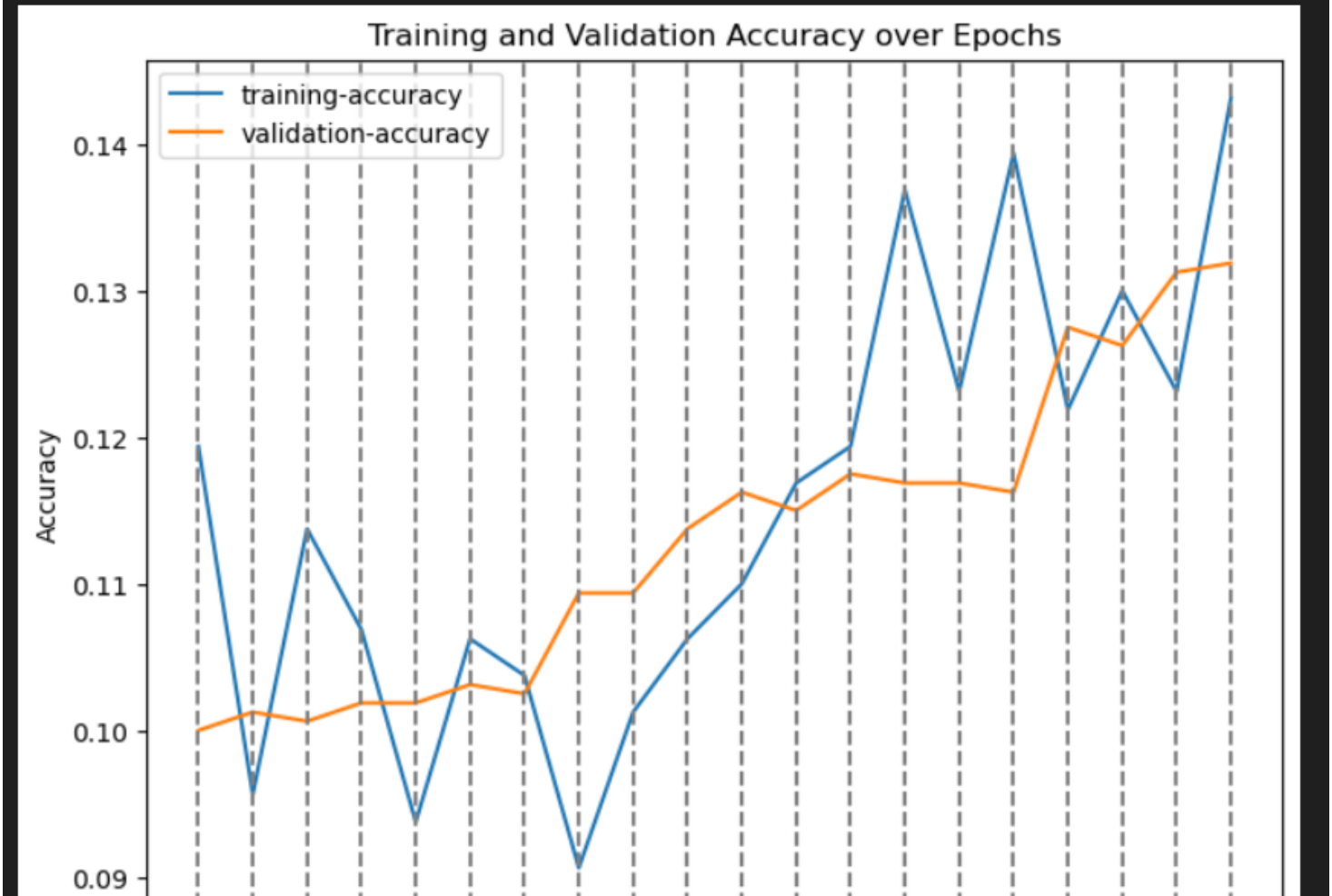
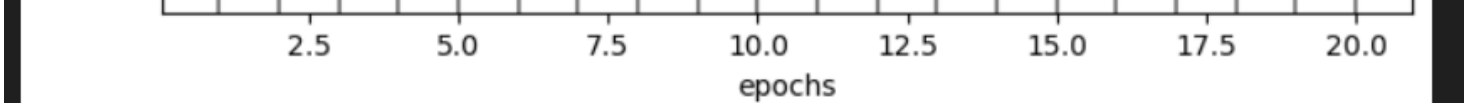
Machine learning Assignment - 4

Part - 3


Training and Validation Loss over Epochs
training-loss
validation-loss
loss
0.230
0.229
0.228
0.227
0.226
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
epochs


Training and Validation Accuracy over Epochs
training-accuracy
validation-accuracy
Accuracy
0.14
0.13
0.12
0.11
0.10
0.09




```
Accuracy for class: plane is 0.0 %
Accuracy for class: car   is 61.6 %
Accuracy for class: bird  is 0.0 %
Accuracy for class: cat   is 0.0 %
Accuracy for class: deer  is 72.6 %
Accuracy for class: dog   is 0.0 %
Accuracy for class: frog  is 0.0 %
Accuracy for class: horse is 0.0 %
Accuracy for class: ship  is 0.0 %
Accuracy for class: truck is 0.0 %
```

The model was taking a lot of time to train therefore i have trained for 5k samples and taken validation over 1k samples

Part 4

4.1 Hyperparameter tuning

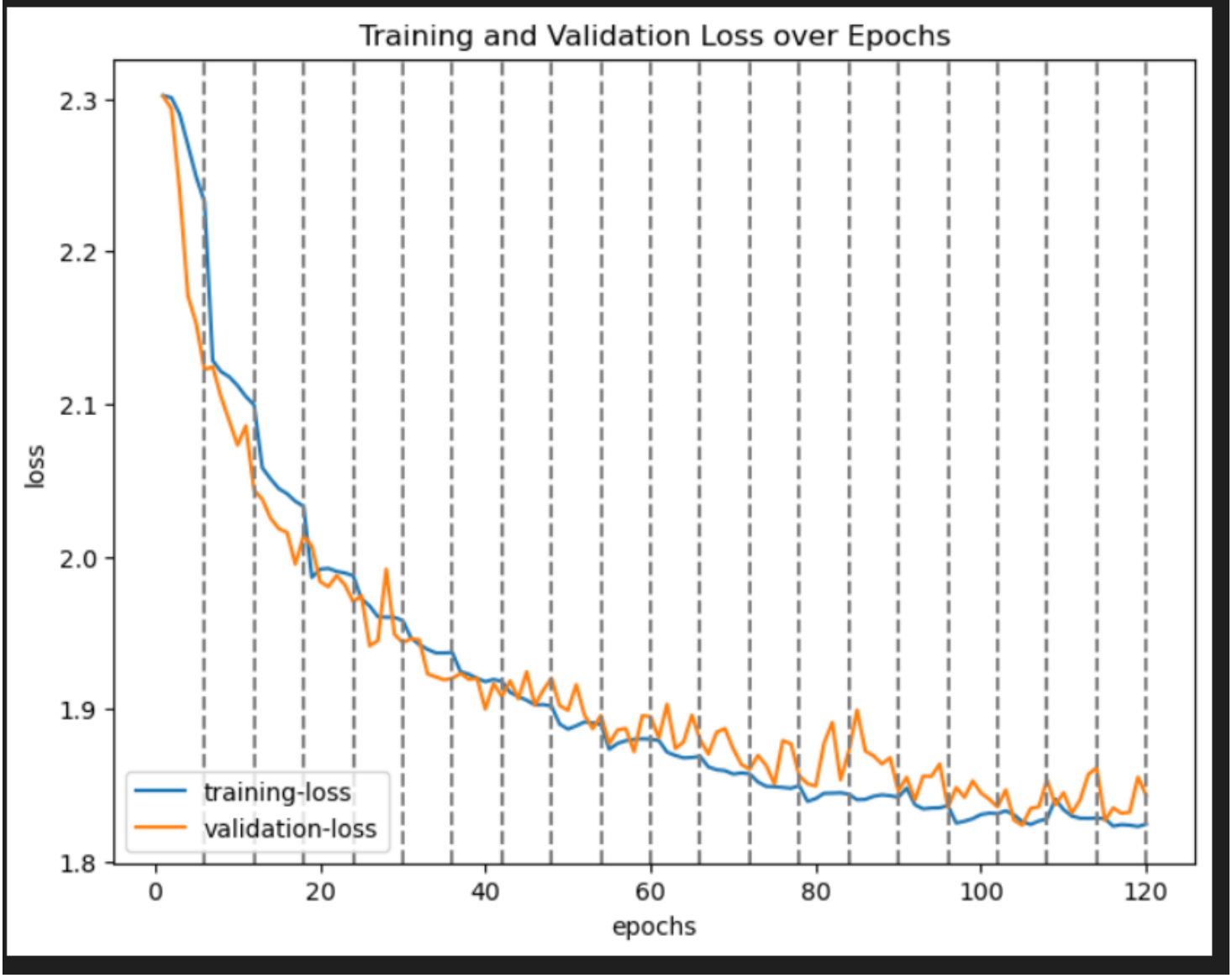
SGD optimizer is used

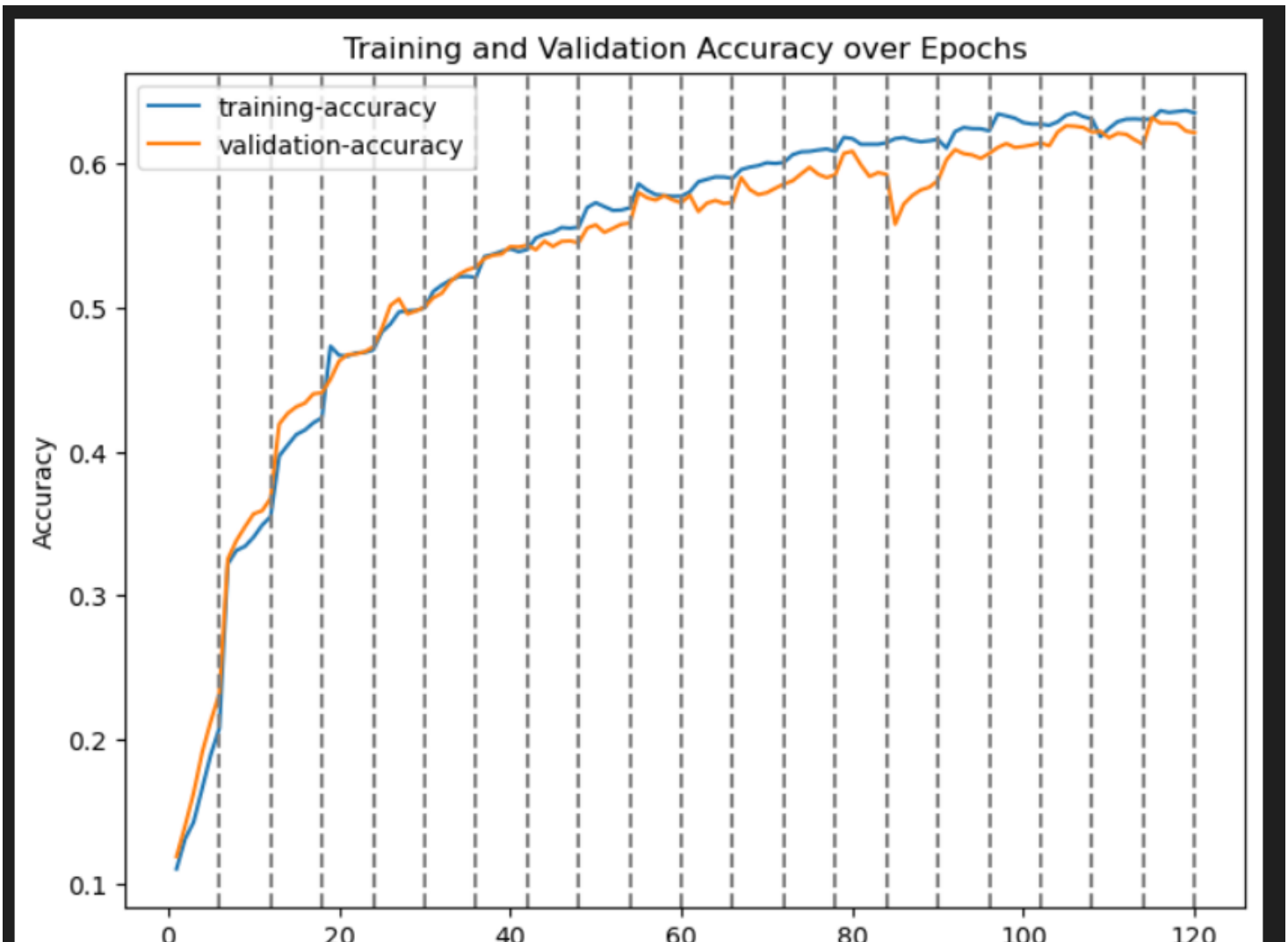
Data augmentation->

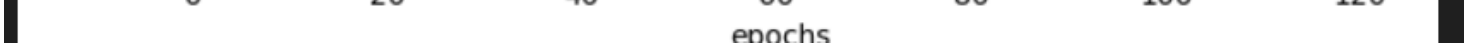
transform = transforms.Compose(
[transforms.ToTensor(),
transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
transforms.RandomHorizontalFlip(p=0.5)
])

Learning Rate

LR = 0.001

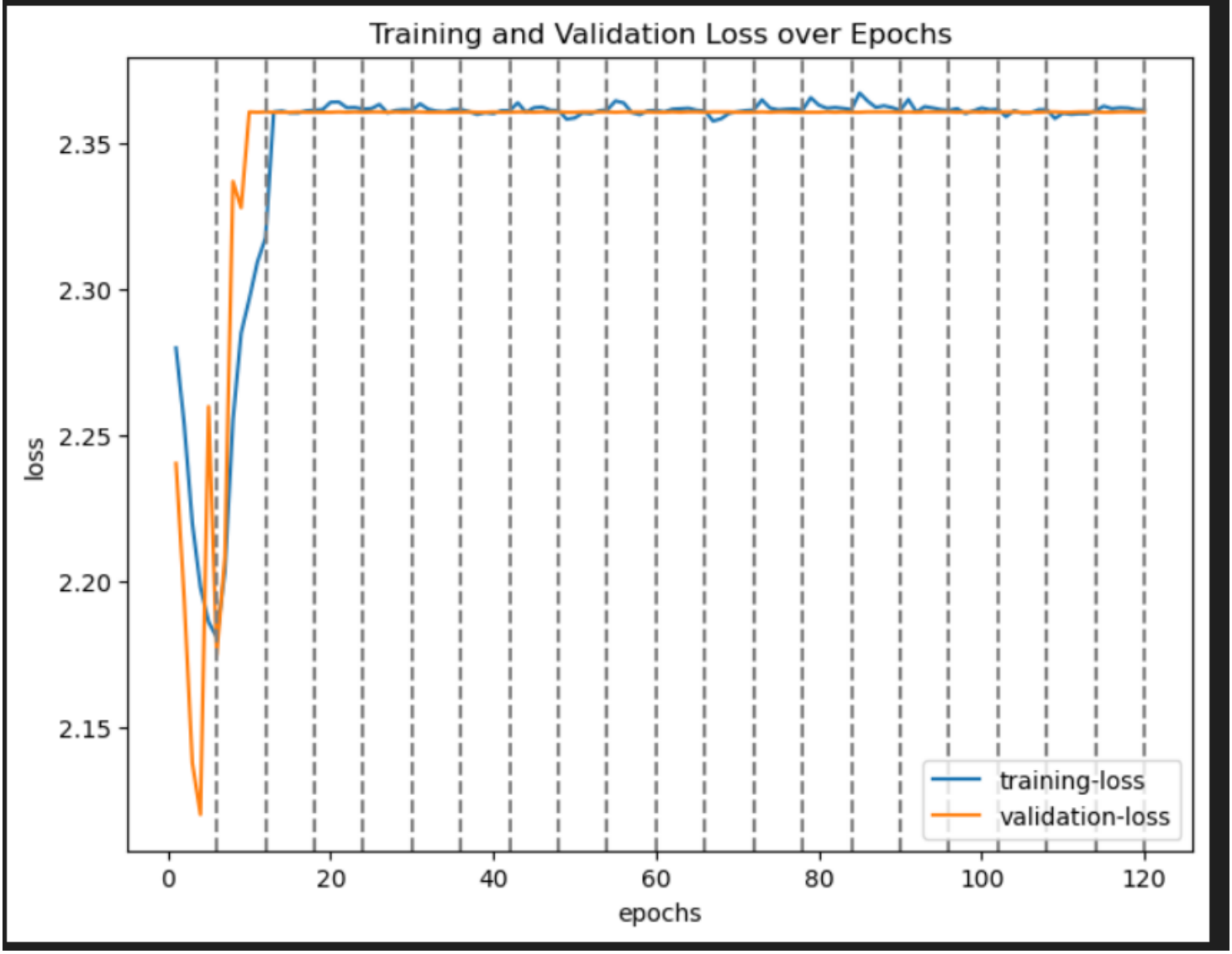

Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.3
2.2
2.1
2.0
1.9
1.8
0
20
40
60
80
100
120

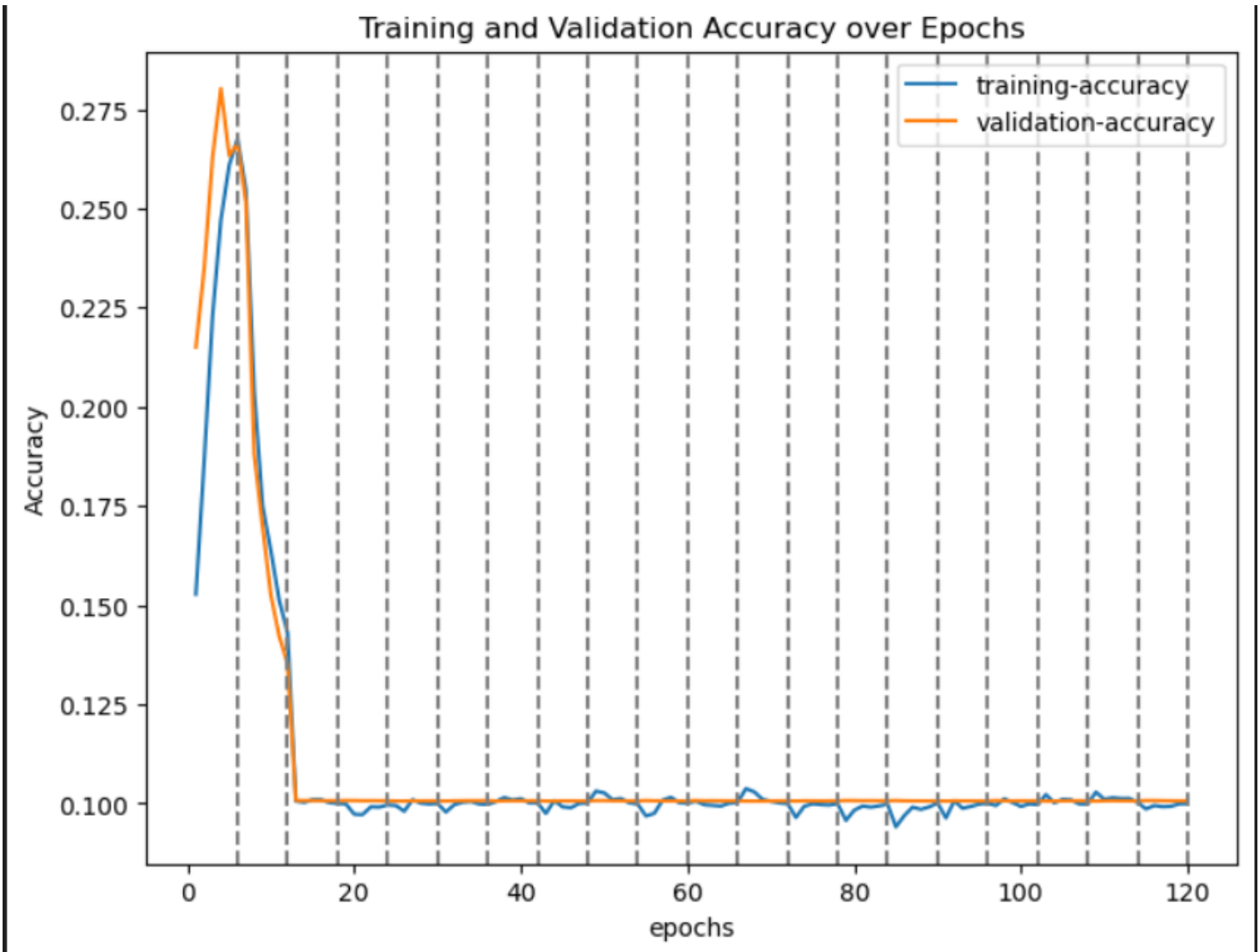

Training and Validation Accuracy over Epochs
Accuracy
training-accuracy
validation-accuracy
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120




```
Accuracy for class: plane is 77.5 %
Accuracy for class: car   is 74.0 %
Accuracy for class: bird  is 0.0 %
Accuracy for class: cat   is 53.7 %
Accuracy for class: deer  is 58.8 %
Accuracy for class: dog   is 46.4 %
Accuracy for class: frog  is 73.6 %
Accuracy for class: horse is 81.6 %
Accuracy for class: ship  is 74.9 %
Accuracy for class: truck is 72.4 %
```

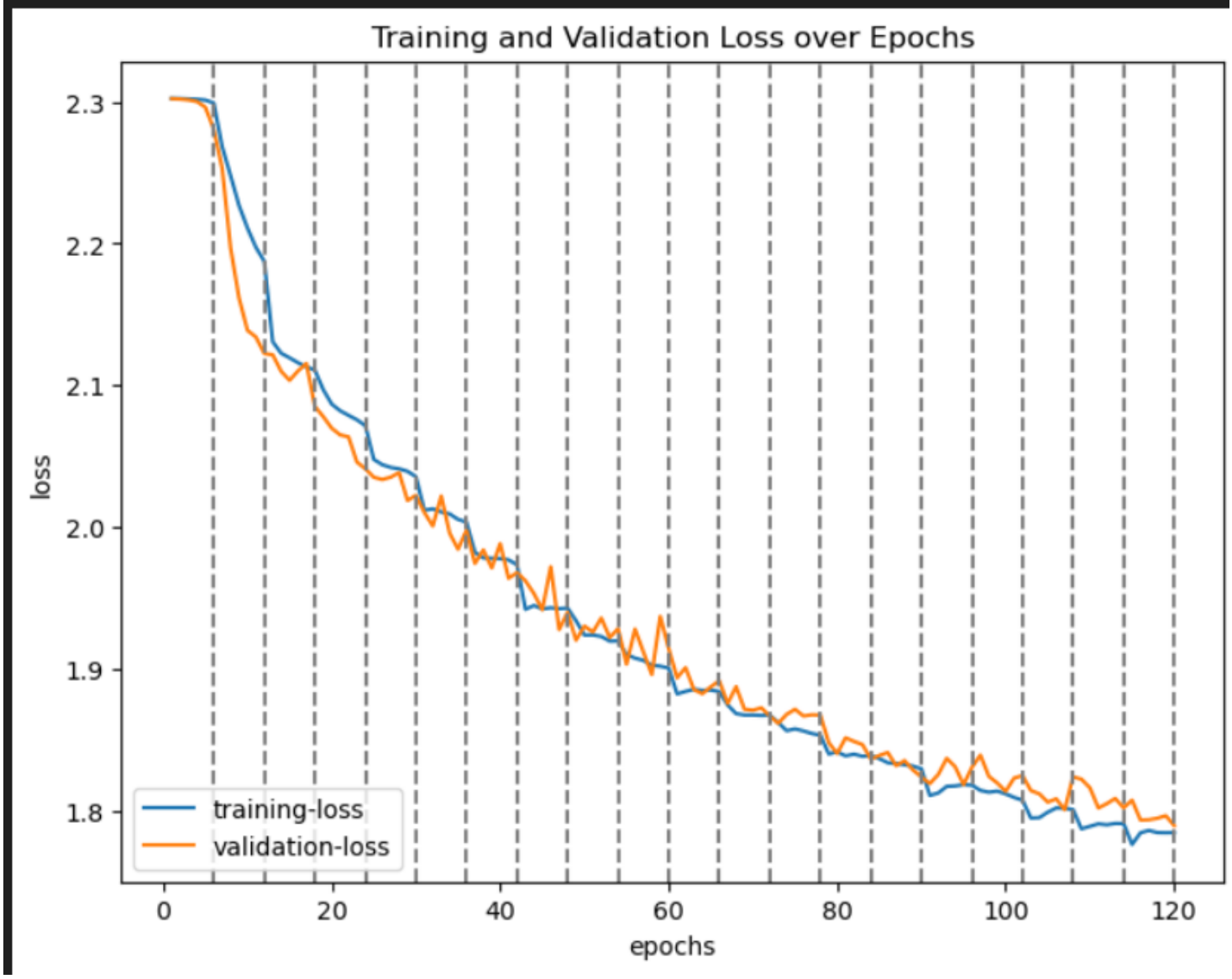
LR = 0.005

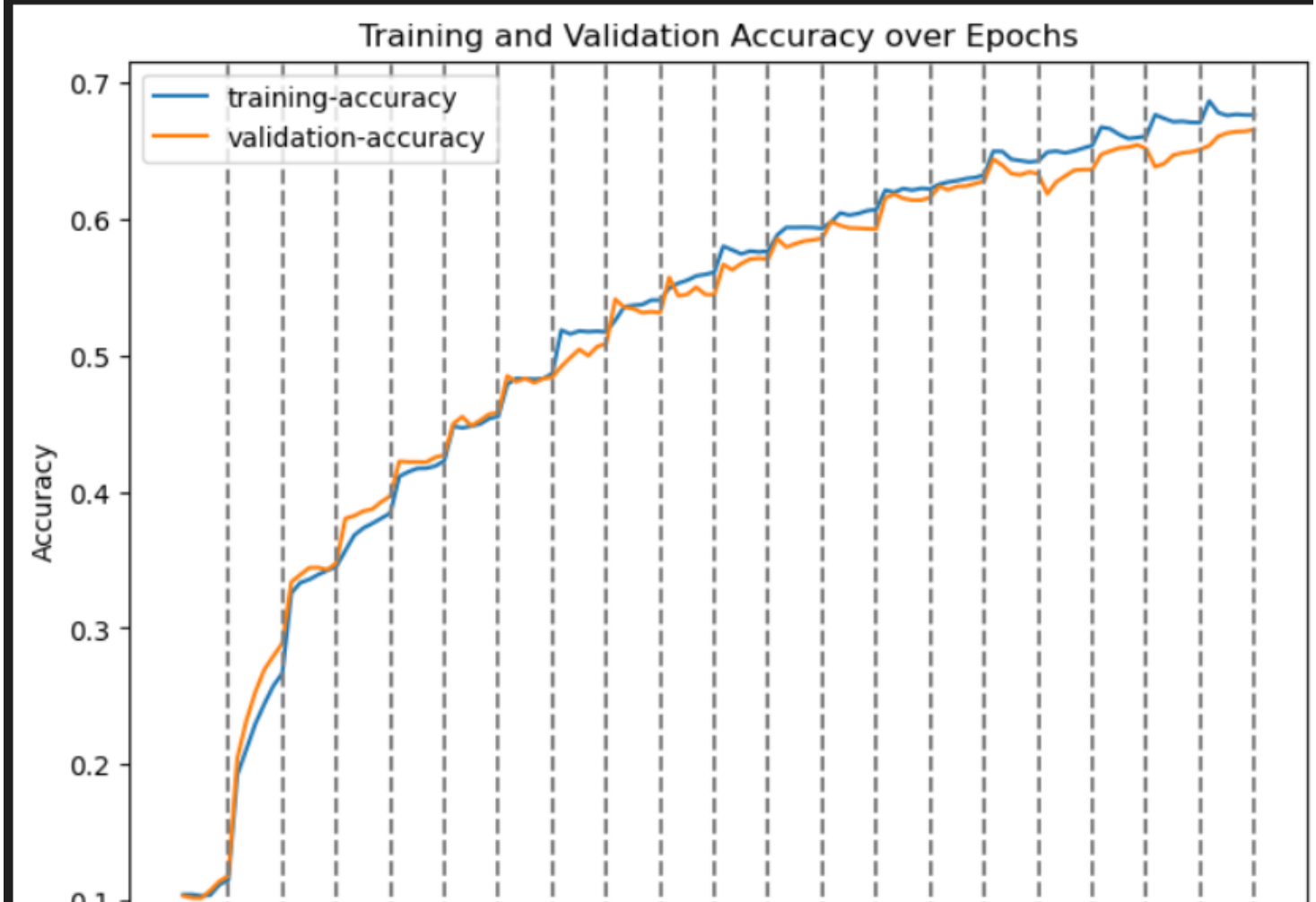

Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.35
2.30
2.25
2.20
2.15
0
20
40
60
80
100
120

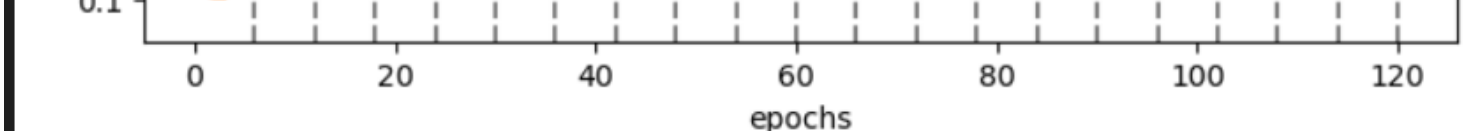

Training and Validation Accuracy over Epochs
Accuracy
epochs
training-accuracy
validation-accuracy
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0.100
0
20
40
60
80
100
120

```
Accuracy for class: plane is 99.9 %
Accuracy for class: car   is 0.0 %
Accuracy for class: bird  is 0.0 %
Accuracy for class: cat   is 0.0 %
Accuracy for class: deer  is 0.0 %
Accuracy for class: dog   is 0.0 %
Accuracy for class: frog  is 0.0 %
Accuracy for class: horse is 0.0 %
Accuracy for class: ship  is 0.0 %
Accuracy for class: truck is 0.0 %
```

LR = 0.0005


Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.3
2.2
2.1
2.0
1.9
1.8
0
20
40
60
80
100
120


Training and Validation Accuracy over Epochs
Accuracy
training-accuracy
validation-accuracy
0.7
0.6
0.5
0.4
0.3
0.2

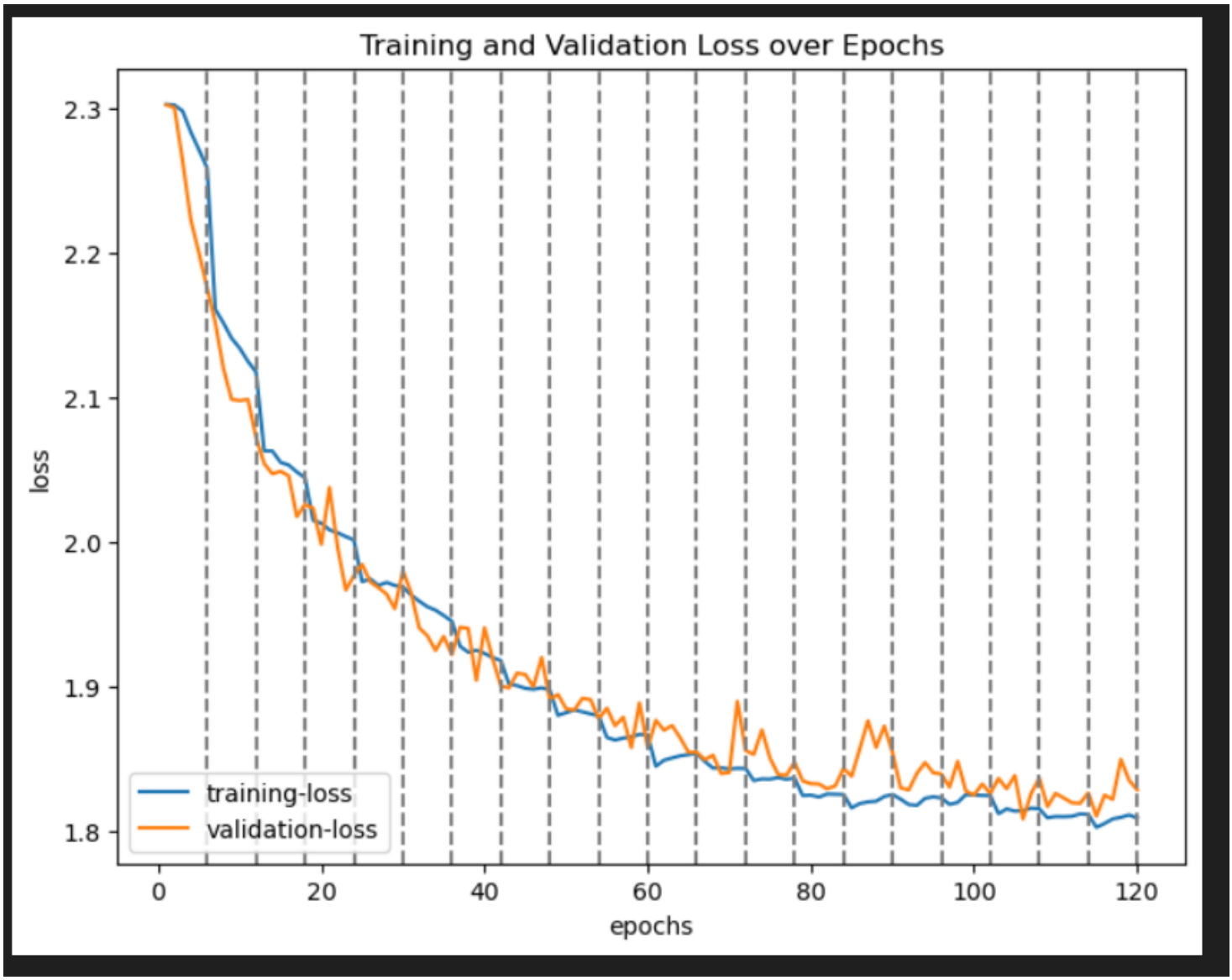


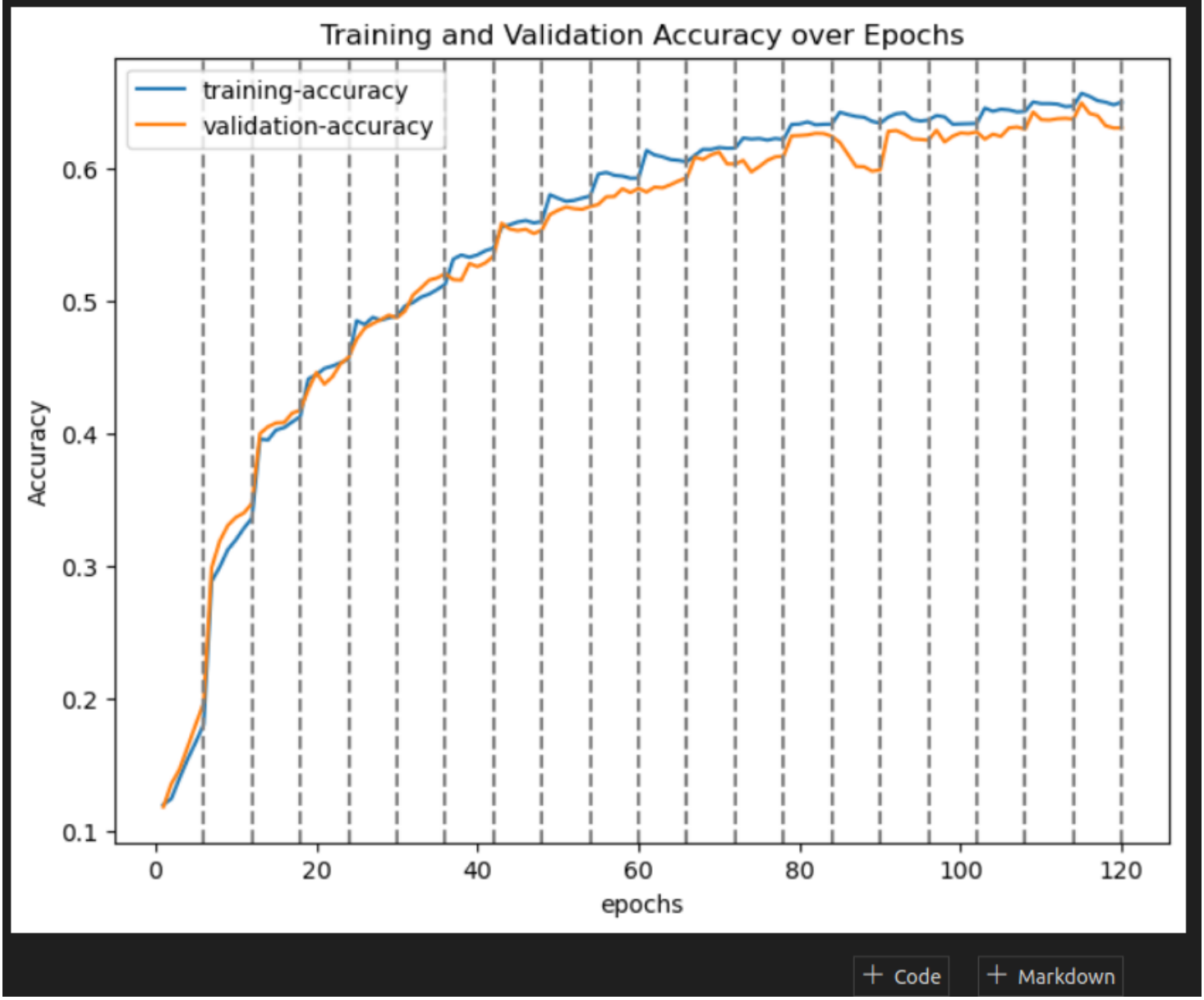

```
Accuracy for class: plane is 63.4 %
Accuracy for class: car   is 80.4 %
Accuracy for class: bird  is 44.0 %
Accuracy for class: cat   is 56.8 %
Accuracy for class: deer  is 52.9 %
Accuracy for class: dog   is 53.8 %
Accuracy for class: frog  is 75.9 %
Accuracy for class: horse is 79.4 %
Accuracy for class: ship  is 76.0 %
Accuracy for class: truck is 79.6 %
```

We can see that the learning rate of 0.005 is bad since the model diverges. Also 0.0005 will take a lot of time to converge therefore 0.001 is the best learning rate

Variation in LR

I have chosen StepLR Scheduler

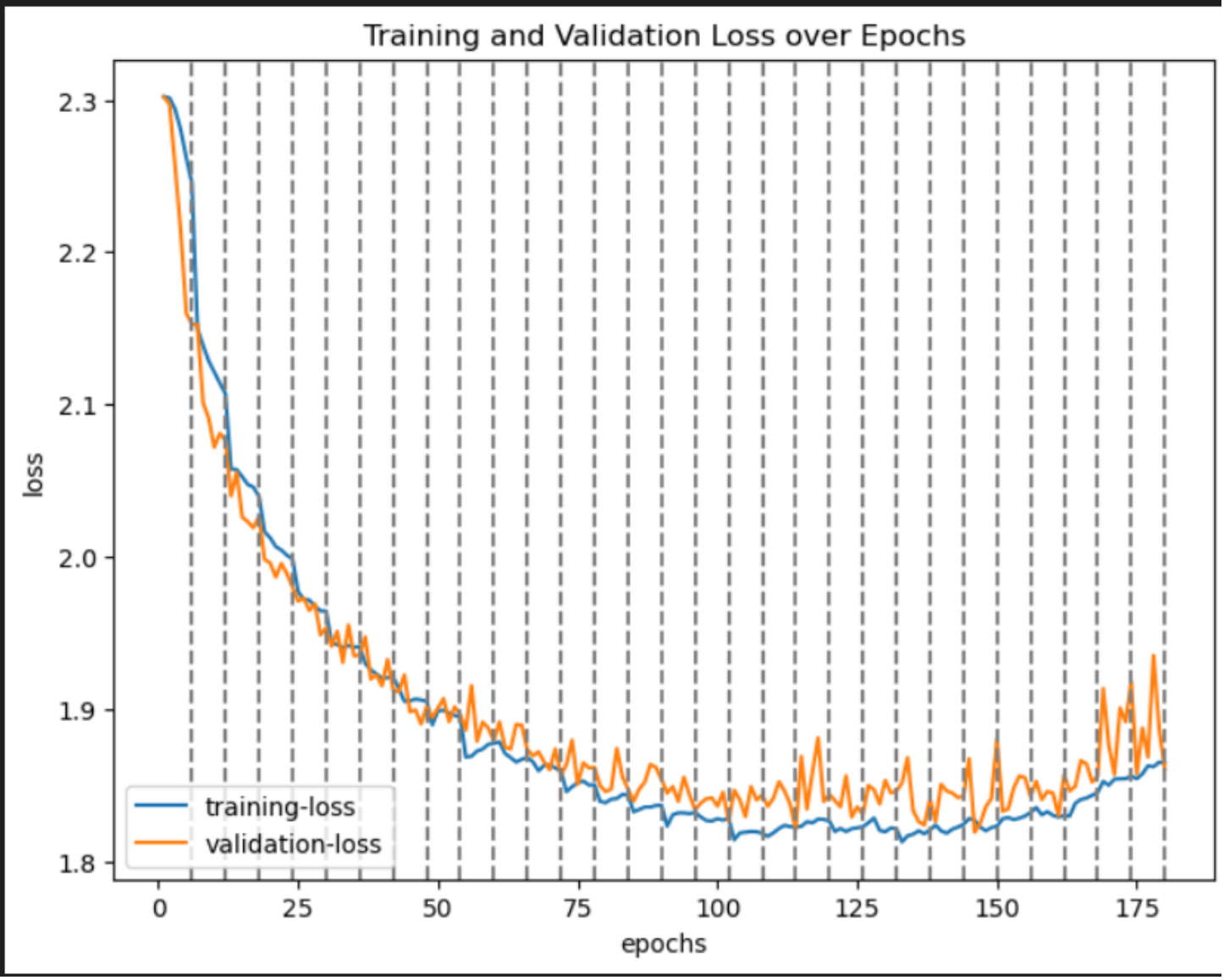


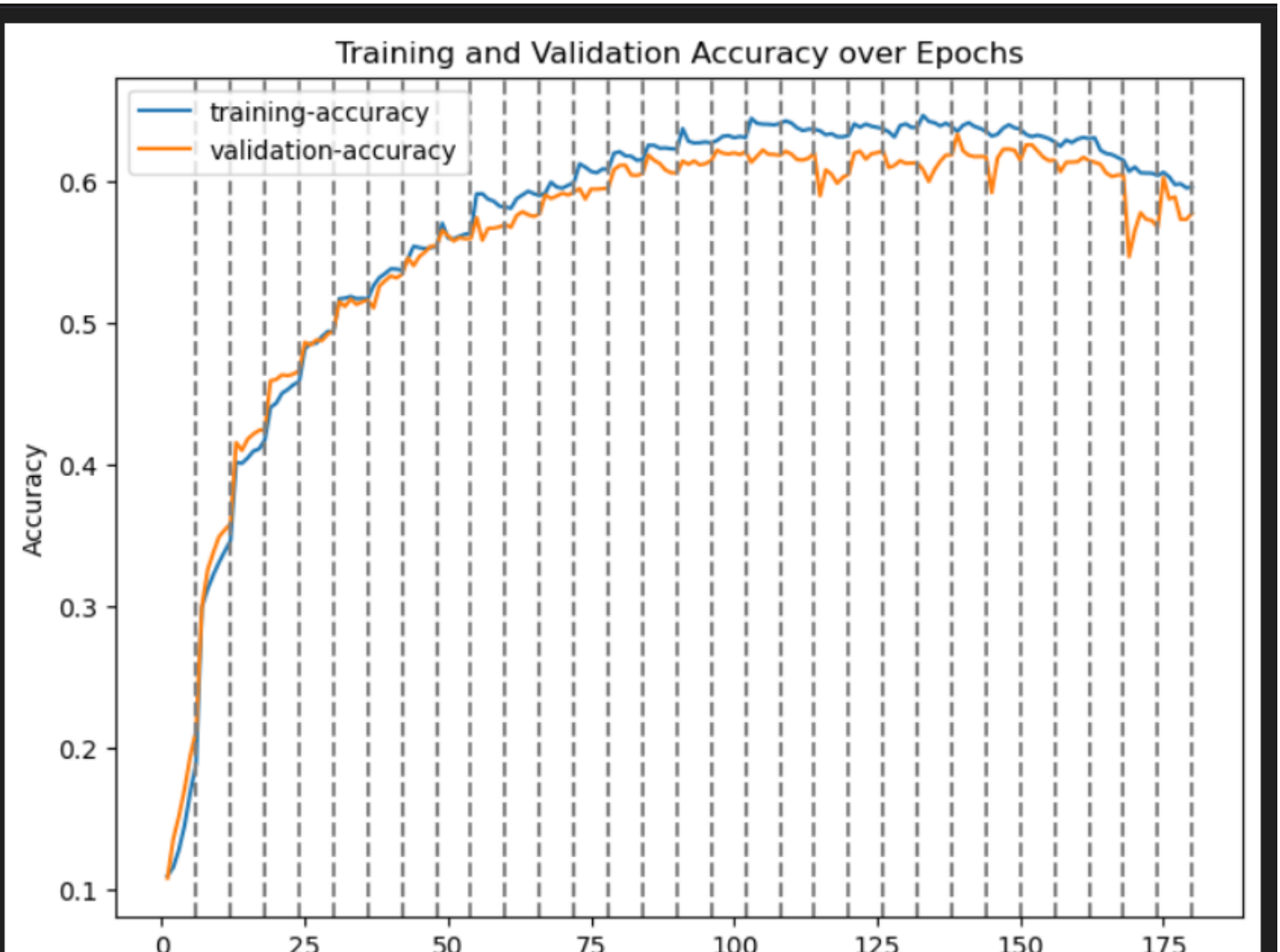


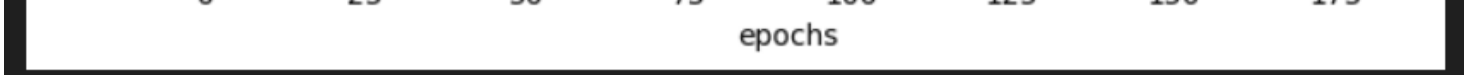

```
Accuracy for class: plane is 67.7 %
Accuracy for class: car   is 61.0 %
Accuracy for class: bird  is 23.9 %
Accuracy for class: cat   is 53.1 %
Accuracy for class: deer  is 74.1 %
Accuracy for class: dog   is 53.2 %
Accuracy for class: frog  is 57.8 %
Accuracy for class: horse is 73.9 %
Accuracy for class: ship  is 58.6 %
Accuracy for class: truck is 62.3 %
```

We can see that the accuracy for some classes have improved a lot. Scheduler helps the error to converge espacially when near to minima. Scheduler is used to improve the performance of a model by adjusting the learning rate over the course of the training process.

Number of epochs


Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.3
2.2
2.1
2.0
1.9
1.8
0
25
50
75
100
125
150
175


Training and Validation Accuracy over Epochs
Accuracy
training-accuracy
validation-accuracy
0.6
0.5
0.4
0.3
0.2
0.1
0
25
50
75
100
125
150
175

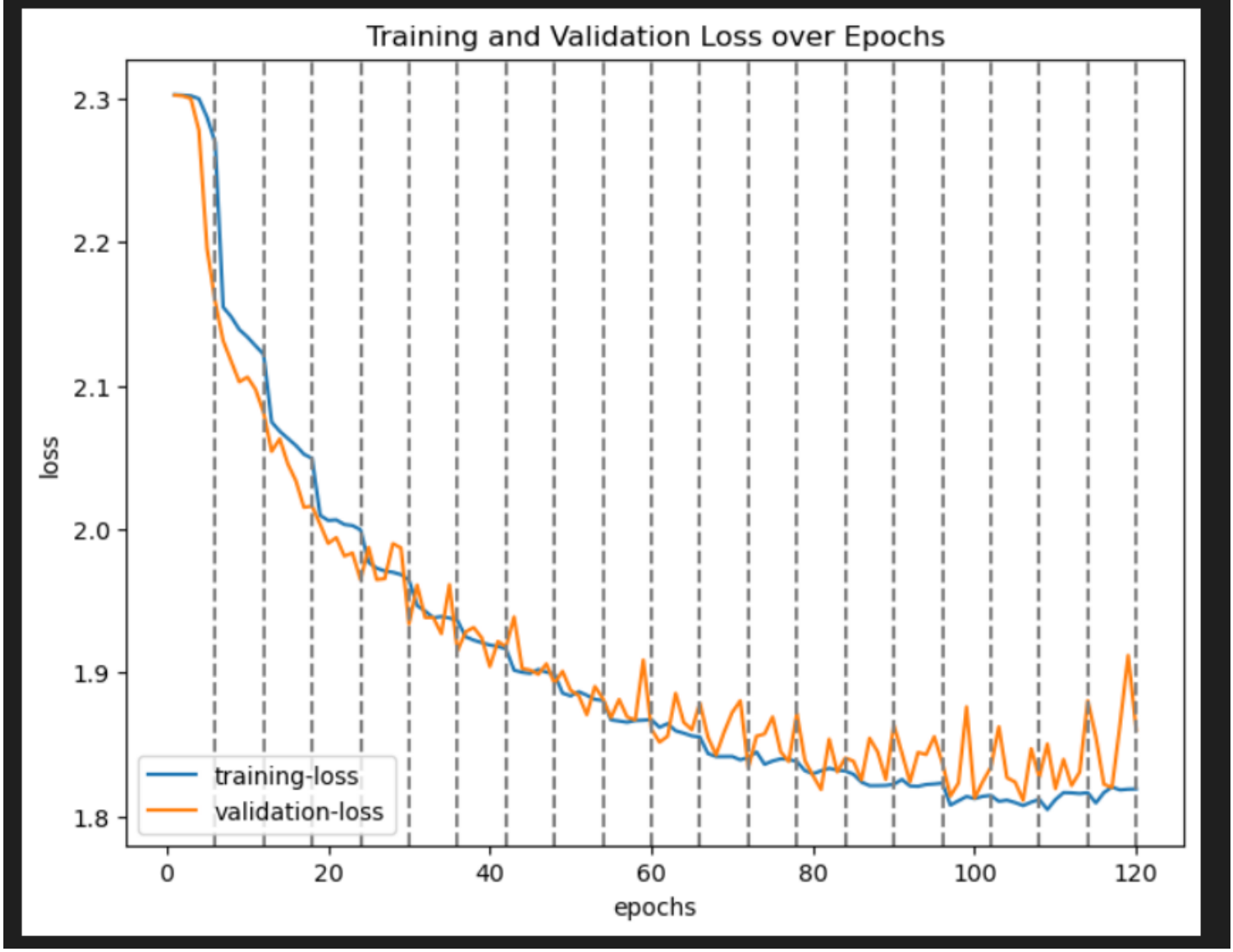


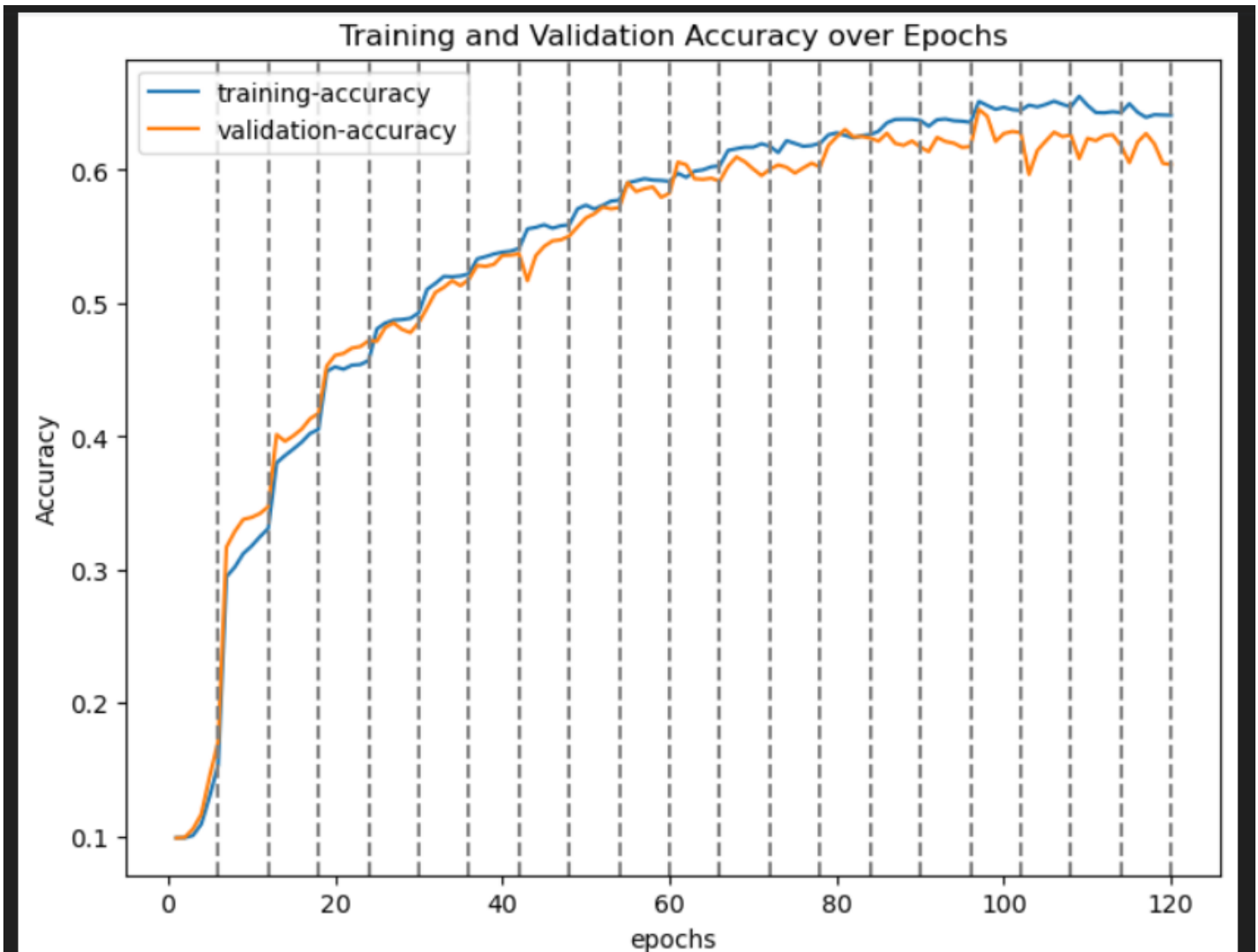

```
Accuracy for class: plane is 69.0 %
Accuracy for class: car   is 73.4 %
Accuracy for class: bird  is 0.0 %
Accuracy for class: cat   is 57.9 %
Accuracy for class: deer  is 64.0 %
Accuracy for class: dog   is 32.9 %
Accuracy for class: frog  is 86.1 %
Accuracy for class: horse is 66.7 %
Accuracy for class: ship  is 82.4 %
Accuracy for class: truck is 61.1 %
```

Increasing the number of epochs have done no good to the error convergence since after some time it started to diverge. It is better to limit the number of epochs to prevent overfitting

Batch Size

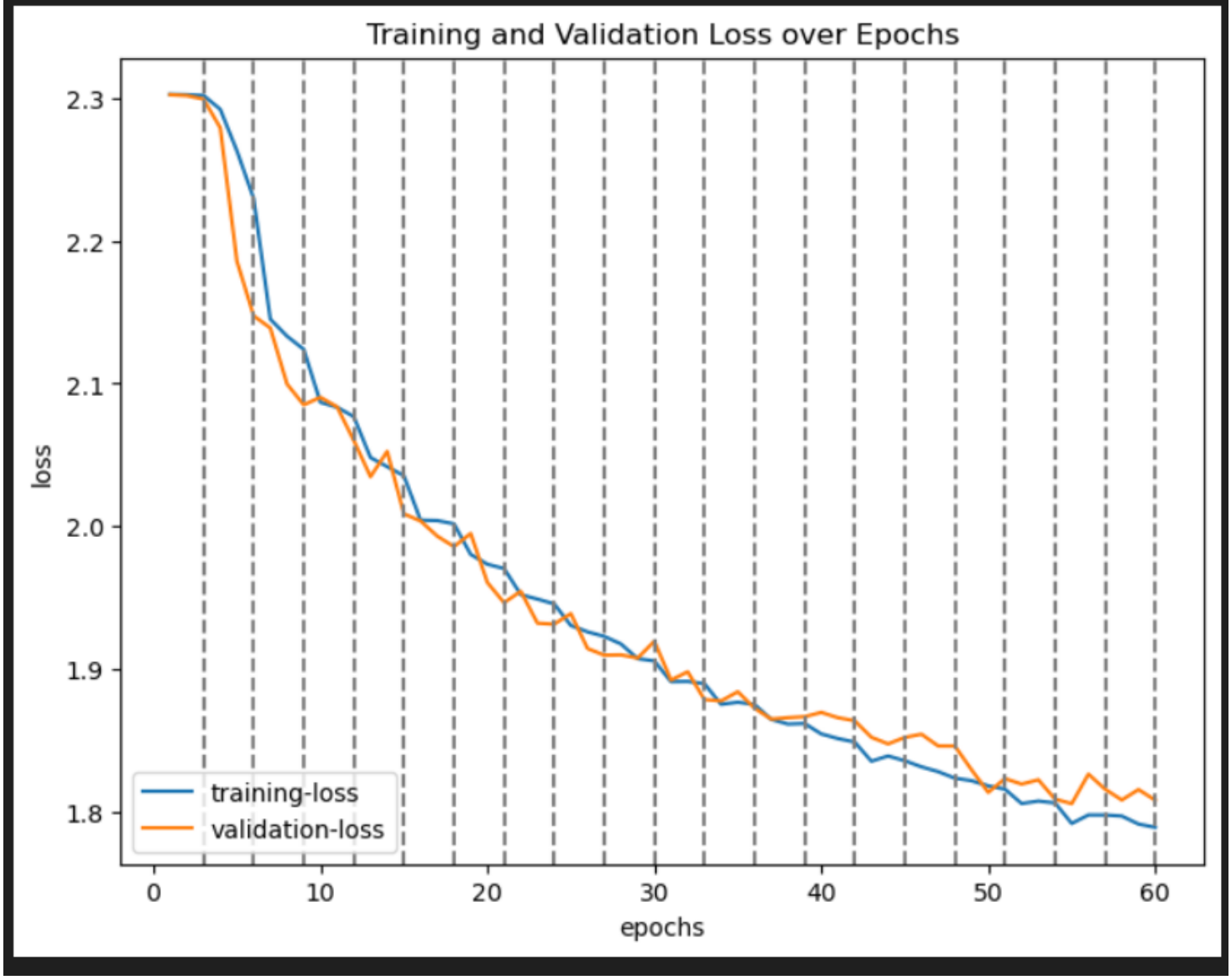
Batch size = 4

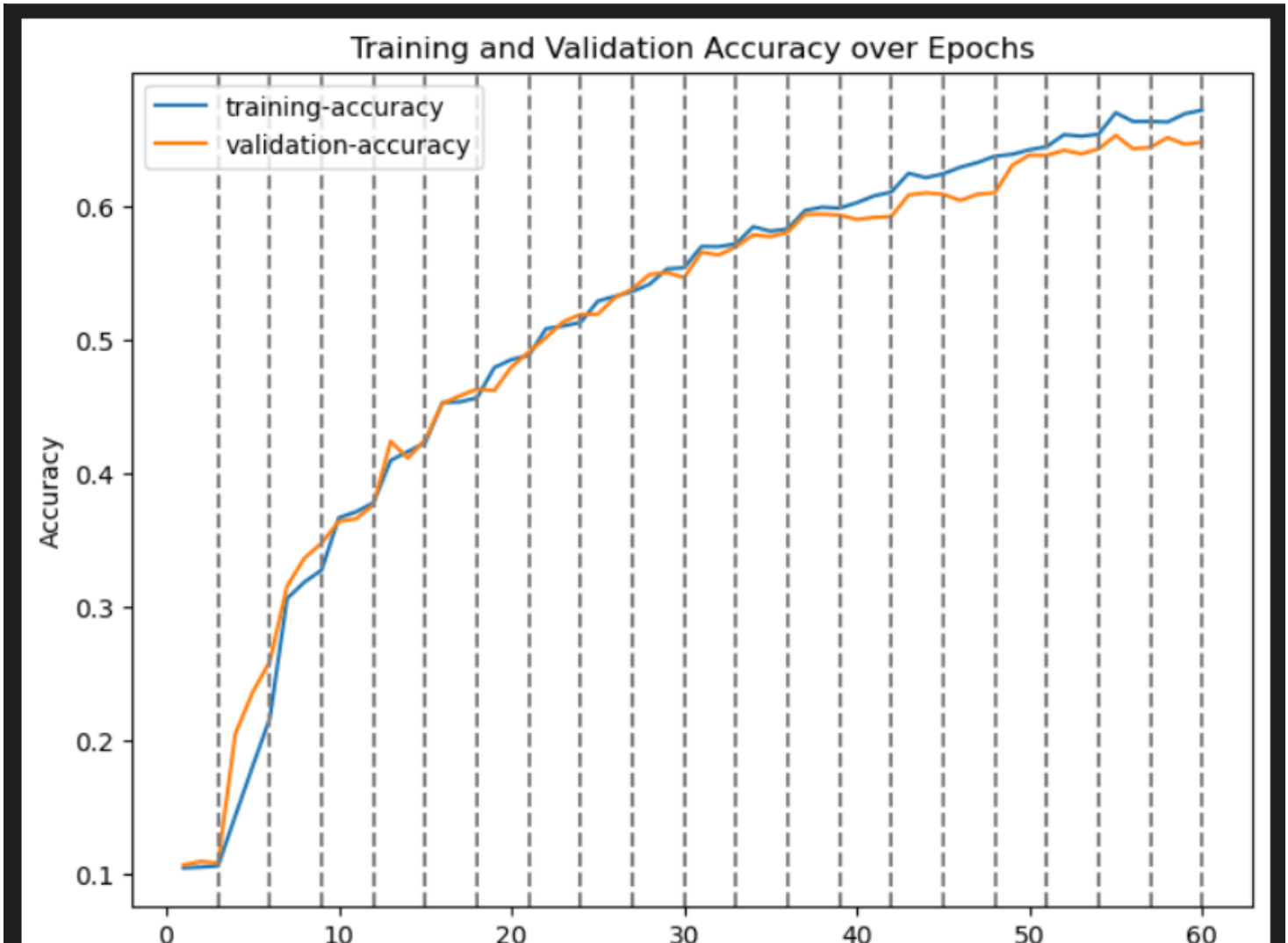

Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.3
2.2
2.1
2.0
1.9
1.8
0
20
40
60
80
100
120

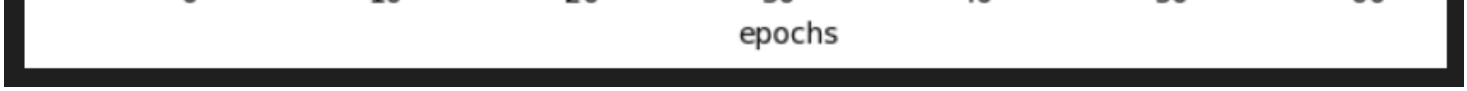

Training and Validation Accuracy over Epochs
Accuracy
epochs
training-accuracy
validation-accuracy
0.6
0.5
0.4
0.3
0.2
0.1
0
20
40
60
80
100
120

```
Accuracy for class: plane is 65.3 %
Accuracy for class: car   is 77.1 %
Accuracy for class: bird  is 55.0 %
Accuracy for class: cat   is 40.9 %
Accuracy for class: deer  is 53.6 %
Accuracy for class: dog   is 62.5 %
Accuracy for class: frog  is 85.7 %
Accuracy for class: horse is 61.4 %
Accuracy for class: ship  is 57.9 %
Accuracy for class: truck is 66.1 %
```

Batch size = 8

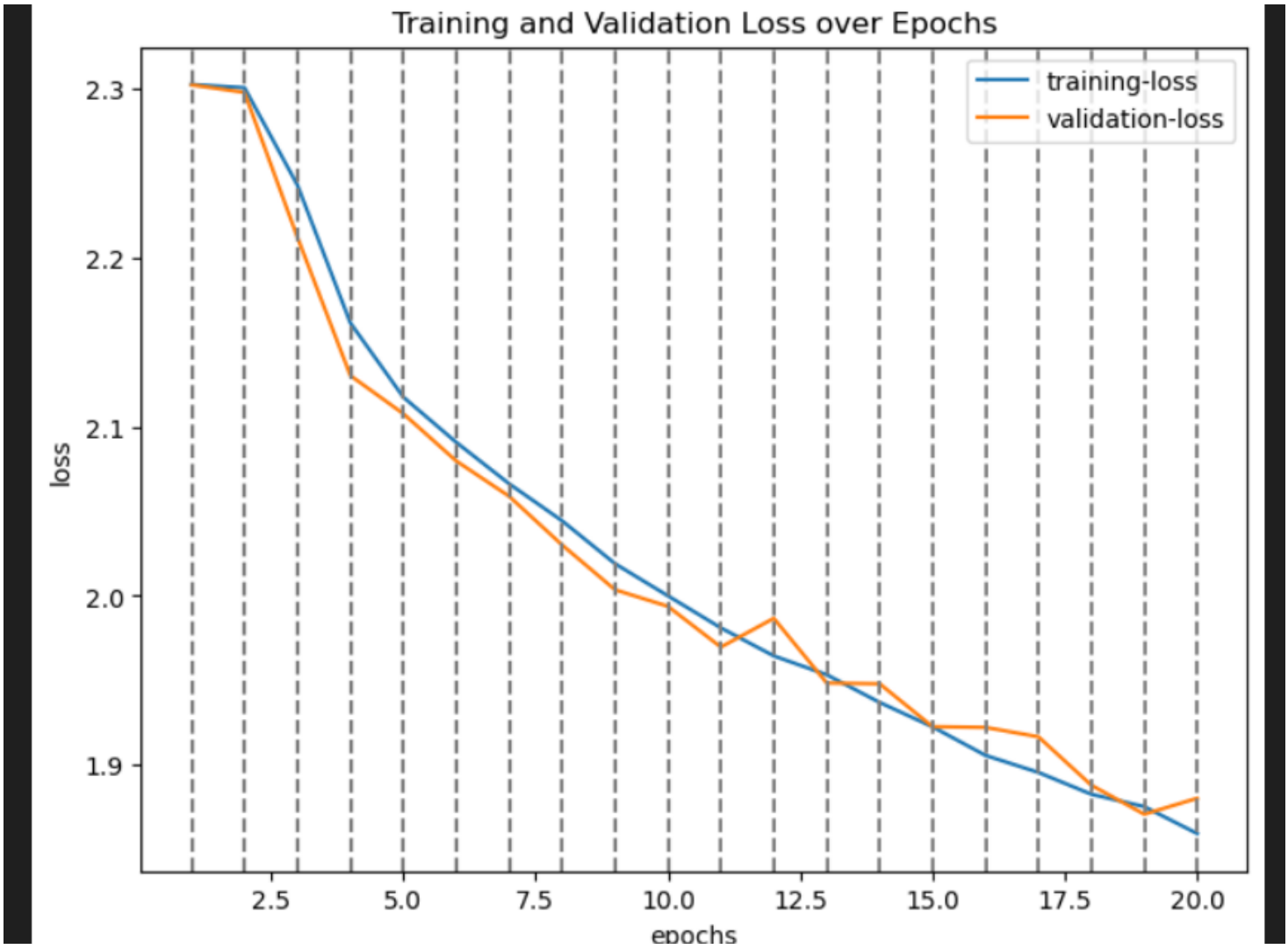

Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.3
2.2
2.1
2.0
1.9
1.8
0
10
20
30
40
50
60

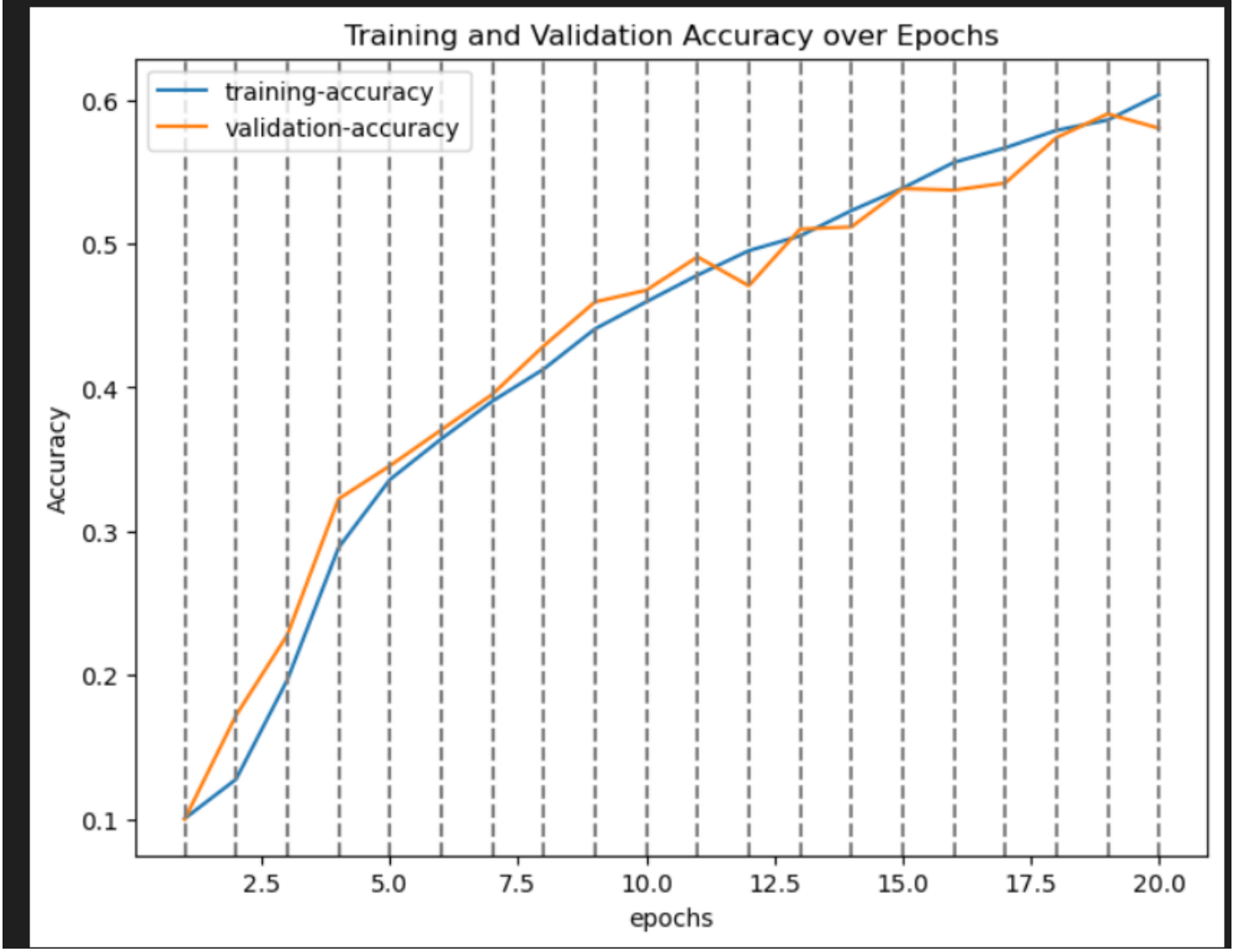

Training and Validation Accuracy over Epochs
Accuracy
training-accuracy
validation-accuracy
0.6
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
50
60




```
Accuracy for class: plane is 67.3 %
Accuracy for class: car   is 72.0 %
Accuracy for class: bird  is 37.3 %
Accuracy for class: cat   is 47.7 %
Accuracy for class: deer  is 56.7 %
Accuracy for class: dog   is 50.7 %
Accuracy for class: frog  is 64.6 %
Accuracy for class: horse is 82.6 %
Accuracy for class: ship  is 76.2 %
Accuracy for class: truck is 83.0 %
```

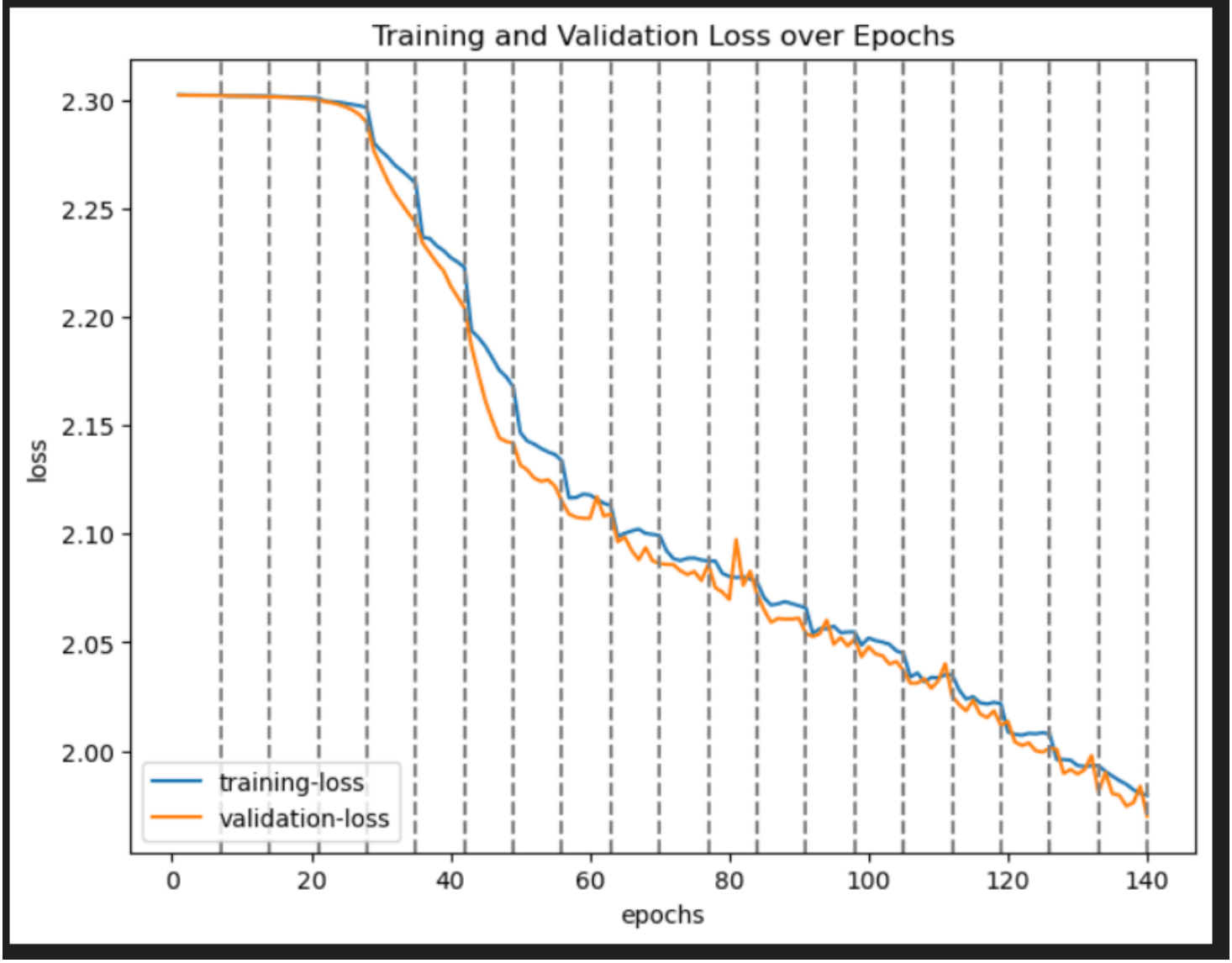
Batch size = 16

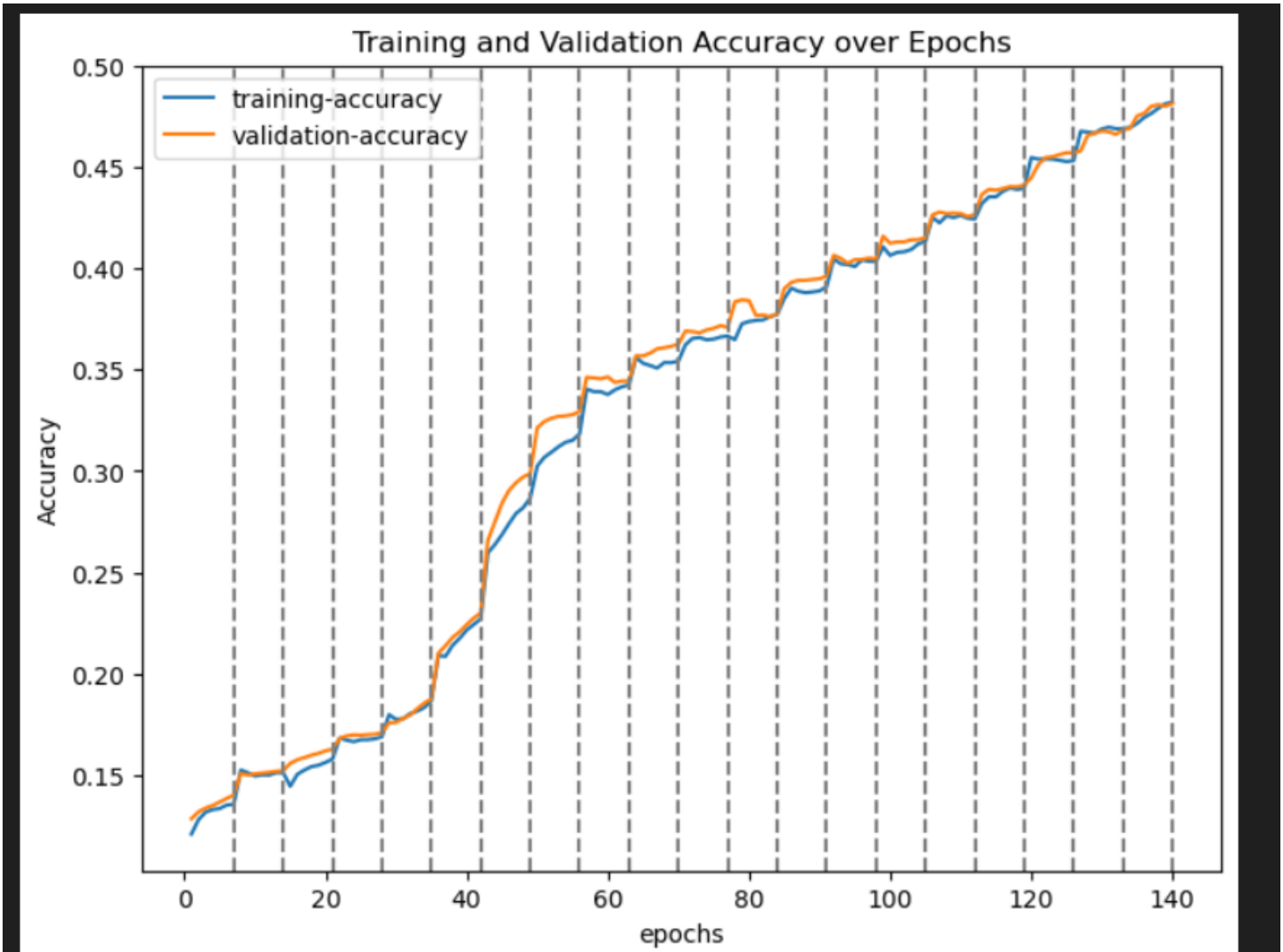





```
Accuracy for class: plane is 63.3 %
Accuracy for class: car   is 66.9 %
Accuracy for class: bird  is 39.9 %
Accuracy for class: cat   is 30.5 %
Accuracy for class: deer  is 49.6 %
Accuracy for class: dog   is 60.0 %
Accuracy for class: frog  is 79.1 %
Accuracy for class: horse is 59.2 %
Accuracy for class: ship  is 73.0 %
Accuracy for class: truck is 76.7 %
```

Batch size = 32


Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.30
2.25
2.20
2.15
2.10
2.05
2.00
0
20
40
60
80
100
120
140

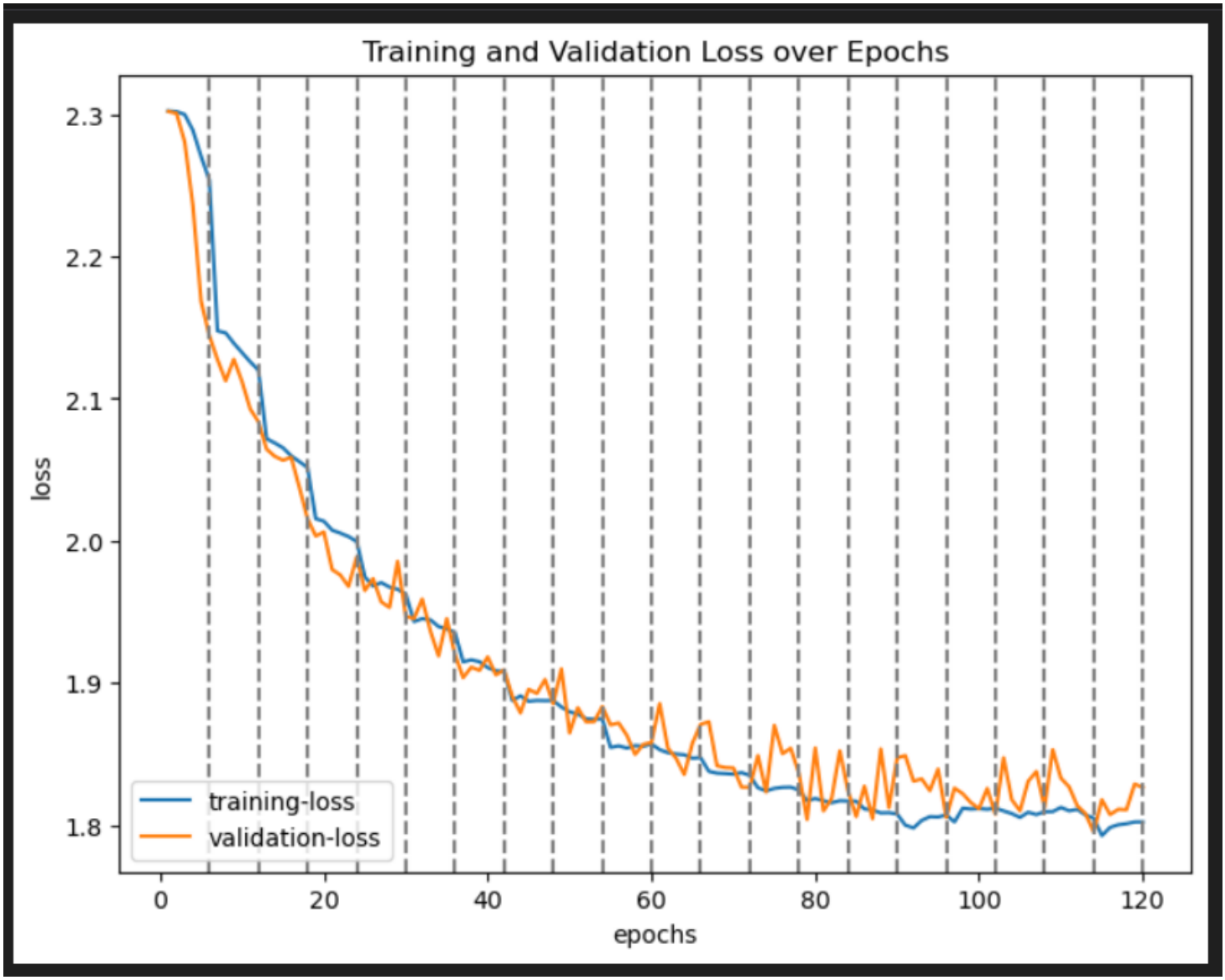

Training and Validation Accuracy over Epochs
Accuracy
epochs
training-accuracy
validation-accuracy
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0
20
40
60
80
100
120
140

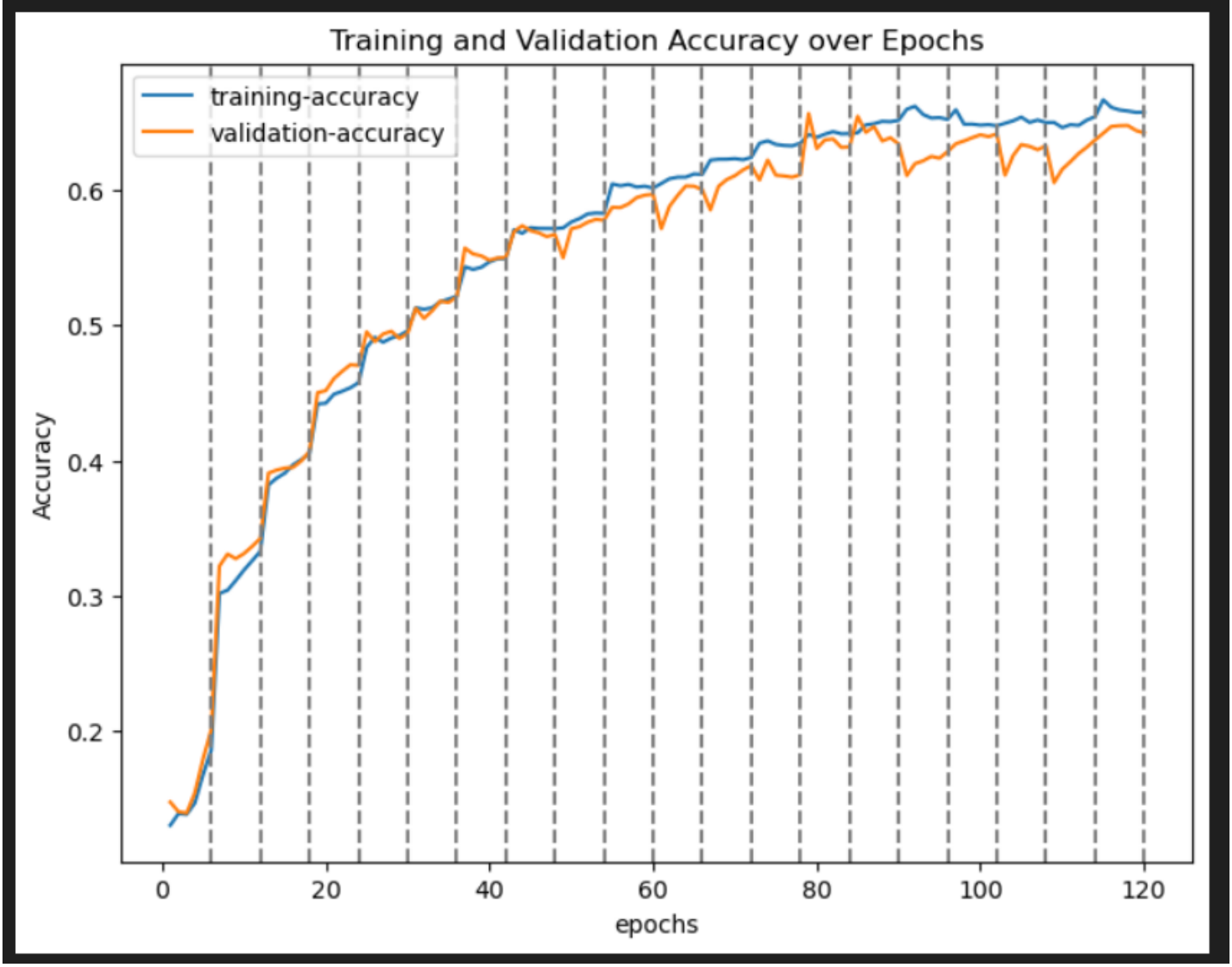
```
Accuracy for class: plane is 61.5 %
Accuracy for class: car   is 66.1 %
Accuracy for class: bird  is 18.6 %
Accuracy for class: cat   is 32.3 %
Accuracy for class: deer  is 32.3 %
Accuracy for class: dog   is 33.4 %
Accuracy for class: frog  is 55.0 %
Accuracy for class: horse is 66.1 %
Accuracy for class: ship  is 44.3 %
Accuracy for class: truck is 68.8 %
```

Observation

We can see that increasing batch size decreases the amount of time the model takes to train but there is a tradeoff between the accuracy. Increasing the batch size may lead to poorer performance. We might need more number of epochs to train.

Loss function - KL Divergence Loss






```
Accuracy for class: plane is 78.4 %
Accuracy for class: car   is 84.1 %
Accuracy for class: bird  is 51.3 %
Accuracy for class: cat   is 52.9 %
Accuracy for class: deer  is 50.0 %
Accuracy for class: dog   is 51.6 %
Accuracy for class: frog  is 69.0 %
Accuracy for class: horse is 68.8 %
Accuracy for class: ship  is 64.5 %
Accuracy for class: truck is 69.1 %
```

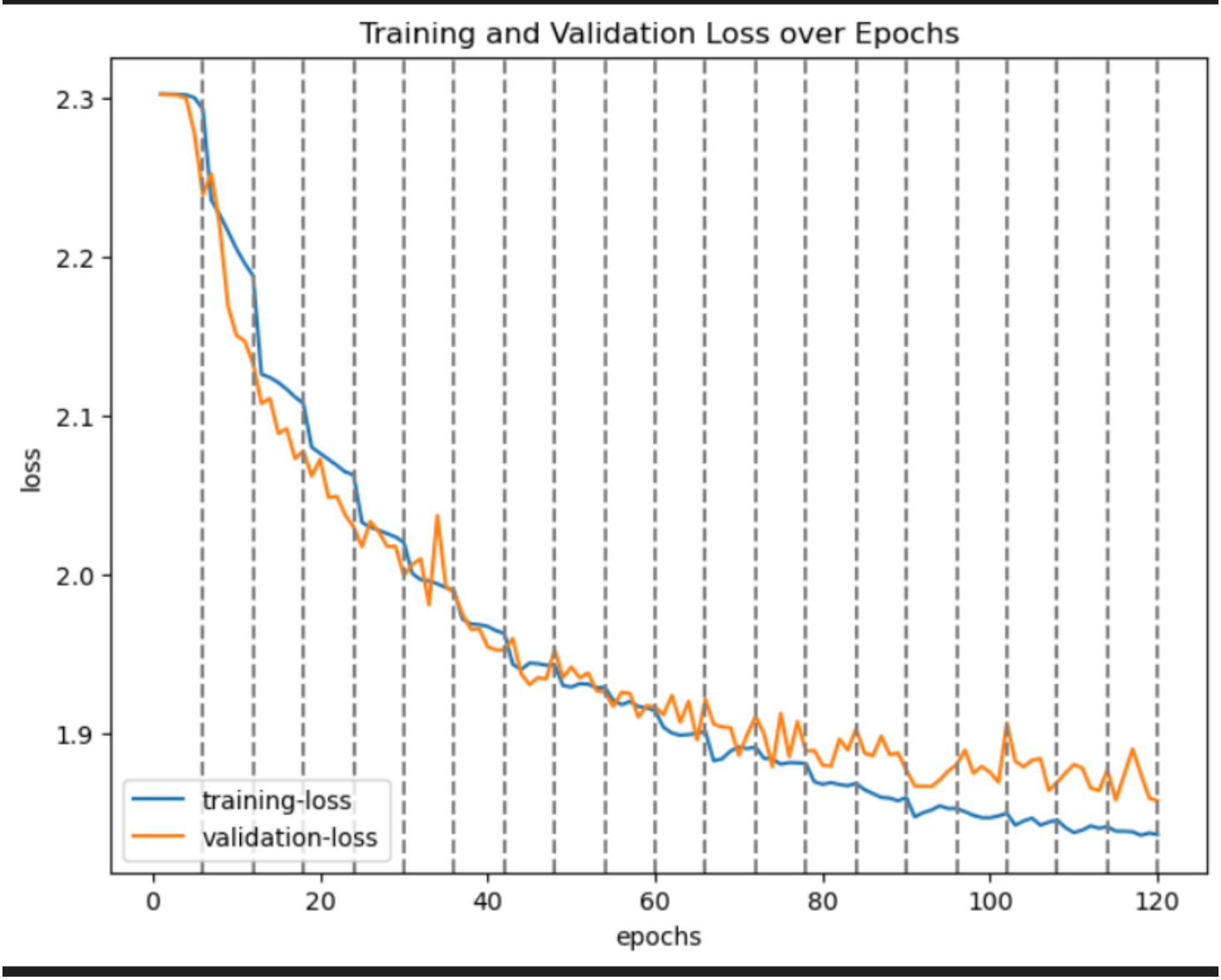
We can see that the model started to recognize some classes. Its accuracy is almost the same but the accuracy for some classes have improved a lot

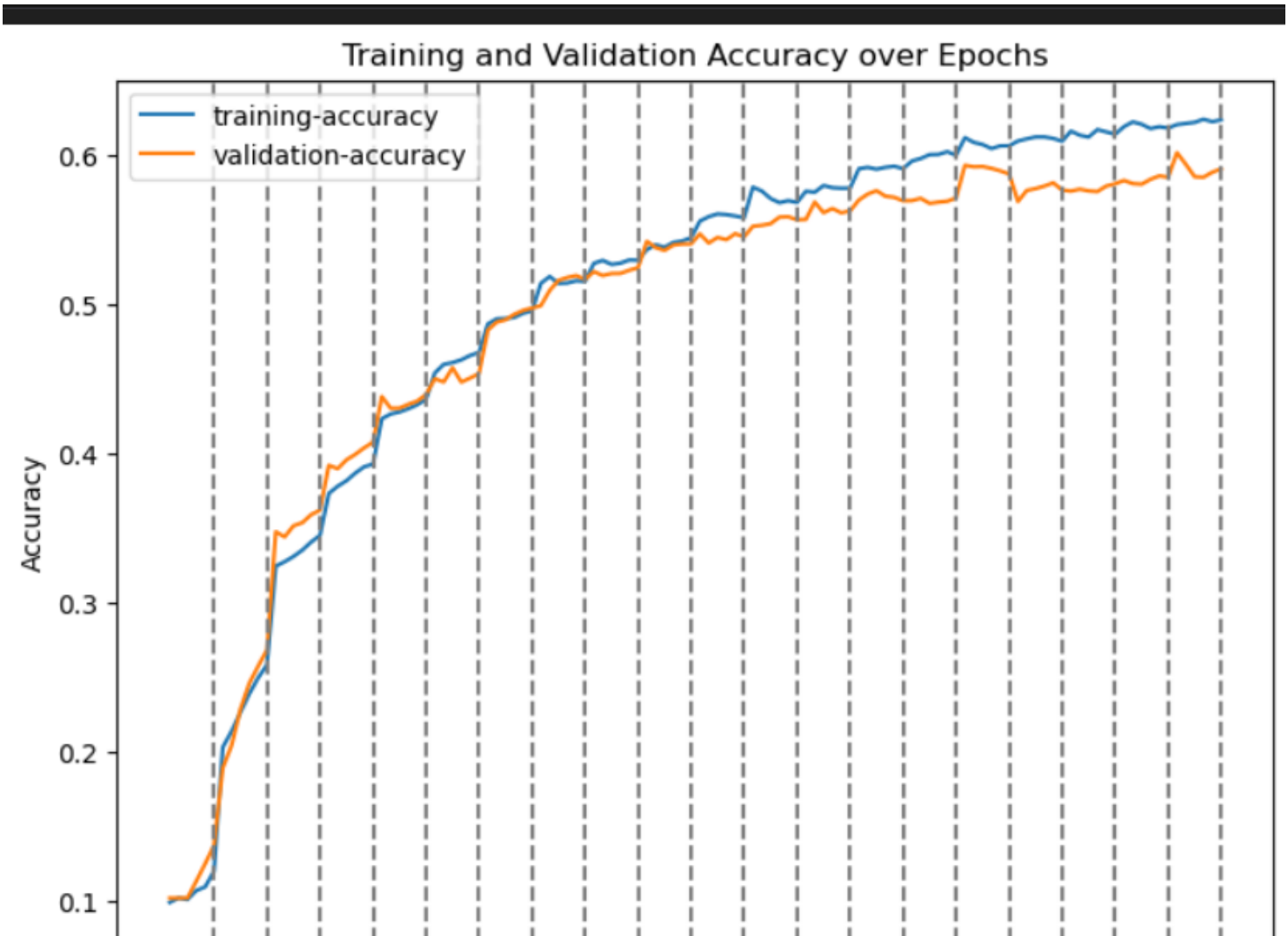
Accuracy for cross entropy for same params

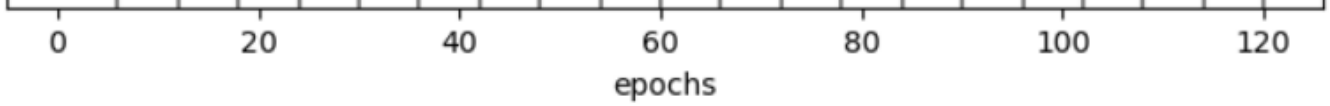
```
Accuracy for class: plane is 77.5 %
Accuracy for class: car   is 74.0 %
Accuracy for class: bird  is 0.0 %
Accuracy for class: cat   is 53.7 %
Accuracy for class: deer  is 58.8 %
Accuracy for class: dog   is 46.4 %
Accuracy for class: frog  is 73.6 %
Accuracy for class: horse is 81.6 %
Accuracy for class: ship  is 74.9 %
Accuracy for class: truck is 72.4 %
```

We can see that model started to recoganize birds with 50%+ accuracy

Data Augmentation ->


Training and Validation Loss over Epochs
loss
epochs
training-loss
validation-loss
2.3
2.2
2.1
2.0
1.9
0
20
40
60
80
100
120


Training and Validation Accuracy over Epochs
Accuracy
training-accuracy
validation-accuracy
0.6
0.5
0.4
0.3
0.2
0.1

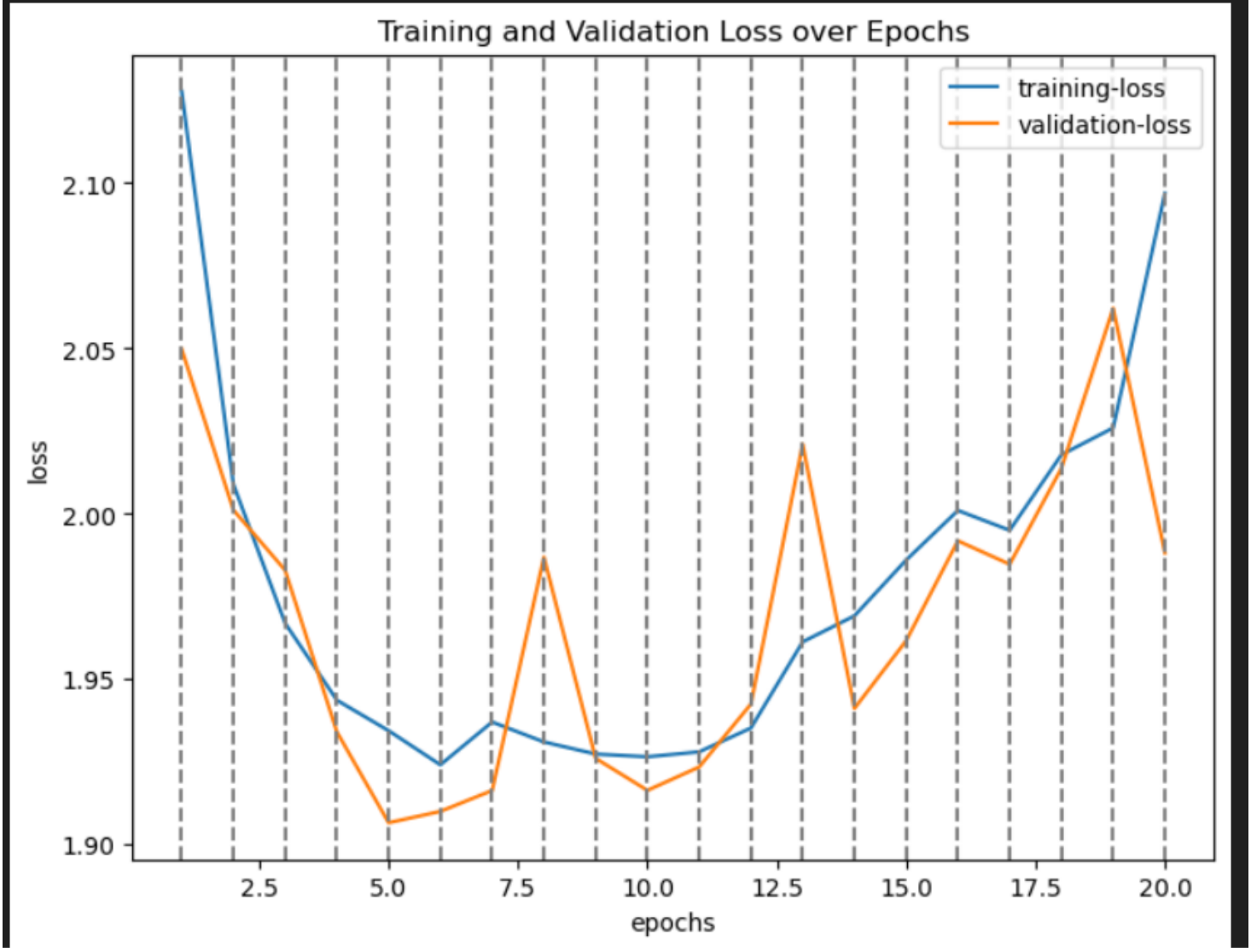


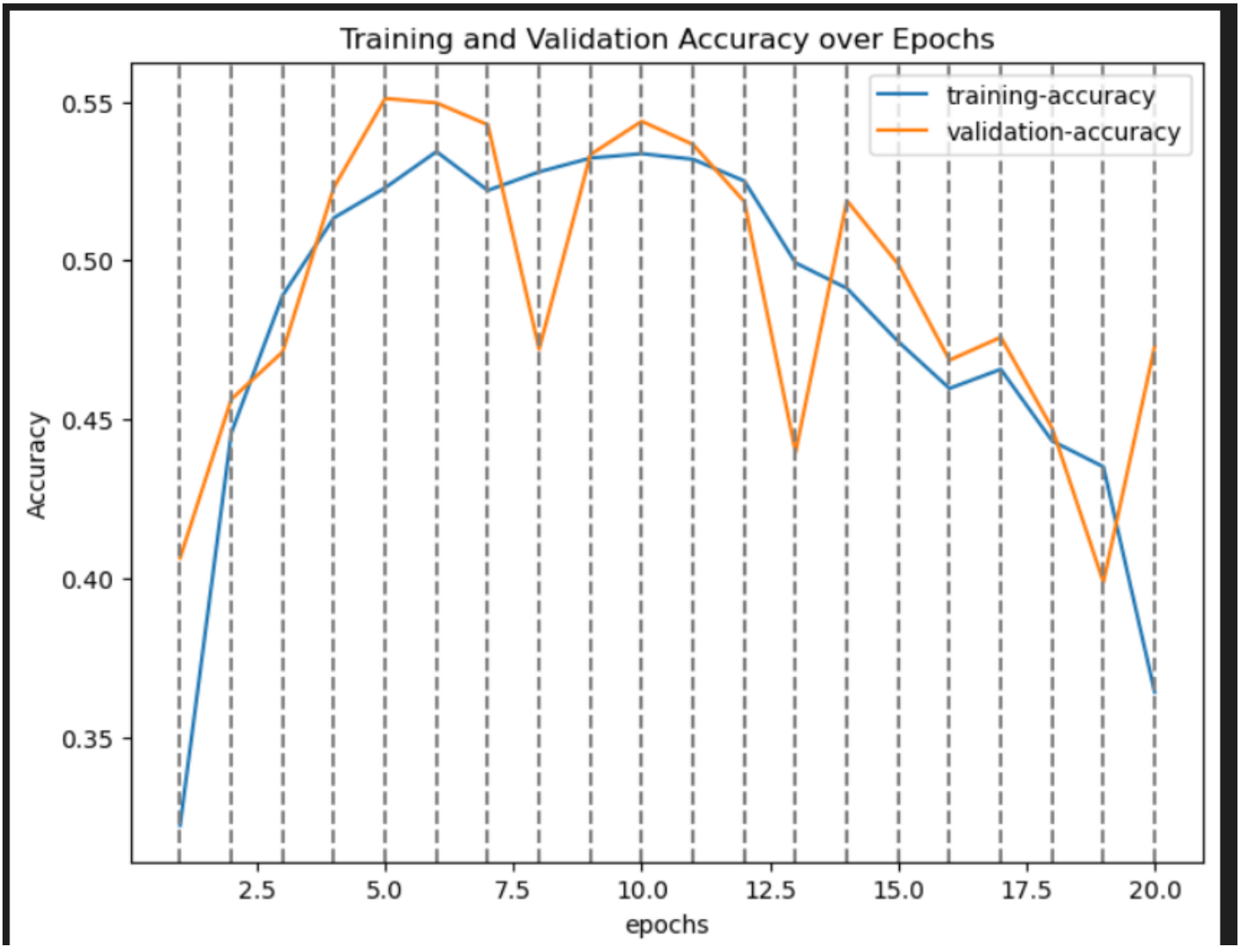

```
Accuracy for class: plane is 62.5 %
Accuracy for class: car   is 72.4 %
Accuracy for class: bird  is 0.0 %
Accuracy for class: cat   is 51.5 %
Accuracy for class: deer  is 57.1 %
Accuracy for class: dog   is 47.9 %
Accuracy for class: frog  is 74.0 %
Accuracy for class: horse is 72.3 %
Accuracy for class: ship  is 88.1 %
Accuracy for class: truck is 75.8 %
```

The model trained trained very fast but its accuracy is little bit poorer than the orignal model.

Adding augmentation is similar to z-transformation. When there are limited number of features, we introduce new features via z-transformation. This is similar to that since we are adding new features by changing the shape/cropping/flpping image etc.

Adam optimizer


Training and Validation Loss over Epochs
training-loss
validation-loss
loss
2.10
2.05
2.00
1.95
1.90
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
epochs


Training and Validation Accuracy over Epochs
training-accuracy
validation-accuracy
Accuracy
0.55
0.50
0.45
0.40
0.35
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
epochs

```
Accuracy for class: plane is 15.0 %
Accuracy for class: car   is 89.2 %
Accuracy for class: bird  is 2.1 %
Accuracy for class: cat   is 1.5 %
Accuracy for class: deer  is 29.7 %
Accuracy for class: dog   is 81.8 %
Accuracy for class: frog  is 39.4 %
Accuracy for class: horse is 15.9 %
Accuracy for class: ship  is 48.4 %
Accuracy for class: truck is 33.5 %
```

We can see that we need to lower the learning rate for Adam optimser to work.

5. Improving the model

data augmentation->

transform = transforms.Compose(
[transforms.ToTensor(),
transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
transforms.RandomHorizontalFlip(p=0.5)
])
new layers
self.conv1 = nn.Conv2d(3, 32, 3)
self.bn1 = nn.BatchNorm2d(32)
self.pool1 = nn.MaxPool2d(2, 2)
self.dropout1 = nn.Dropout2d(0.25)

```
    self.conv2 = nn.Conv2d(32, 64, 5)
    self.bn2 = nn.BatchNorm2d(64)
    self.pool2 = nn.MaxPool2d(2, 2)
    self.dropout2 = nn.Dropout2d(0.25)

    self.conv3 = nn.Conv2d(64, 64, 3)
    self.bn3 = nn.BatchNorm2d(64)

    self.fc1 = nn.Linear(64 * 3 * 3, 64)
    self.dropout3 = nn.Dropout(0.5)
    self.fc2 = nn.Linear(64, 10)
```

BatchNormal2d ->

This layer normalises the data before passing it to the next layer.

Dropout2d ->

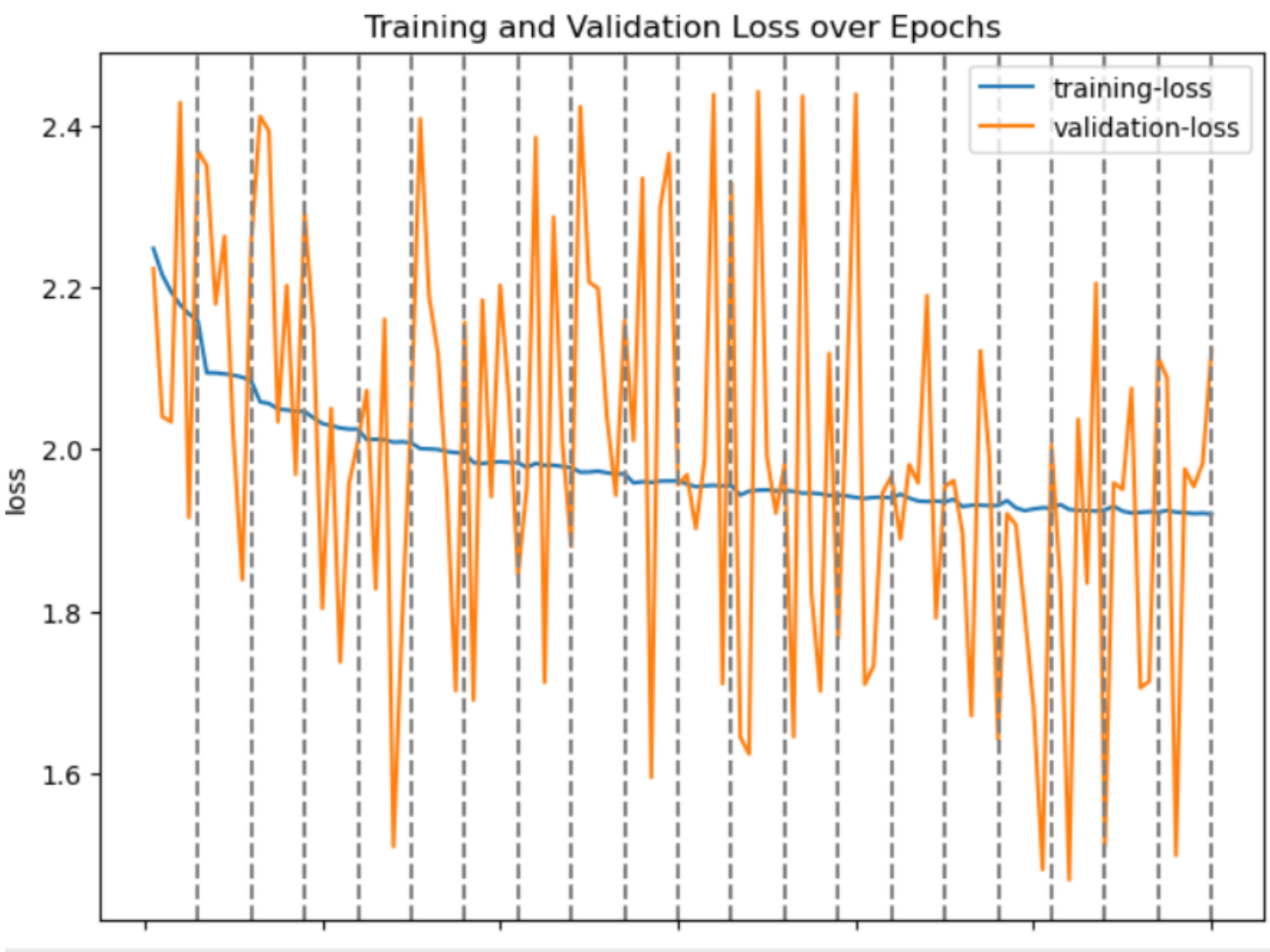
To avoid over fitting I have added another layer that randomly drops out (sets to zero) some of the neurons in a convolutional layer. I helps in regularization.

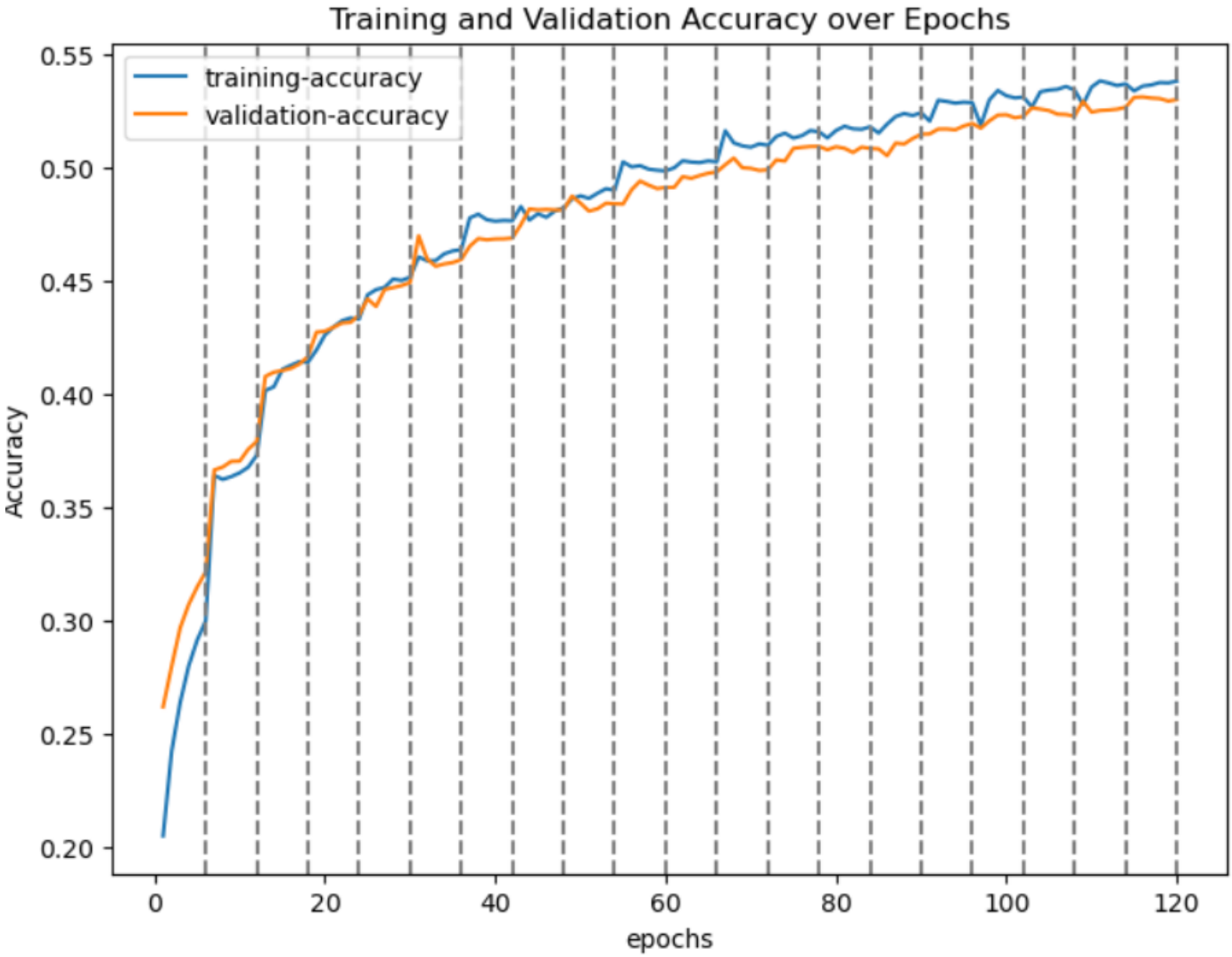
Optimizer ->

I have used adam optimizer

Learning rate ->

I have set the learning rate to 0.0001 since adam optimizer performs better with lower learning rate


Training and Validation Loss over Epochs
training-loss
validation-loss
loss
2.4
2.2
2.0
1.8
1.6


Training and Validation Accuracy over Epochs
training-accuracy
validation-accuracy
Accuracy
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
20
40
60
80
100
120
epochs

```
Accuracy for class: plane is 55.9 %
Accuracy for class: car   is 68.3 %
Accuracy for class: bird  is 37.4 %
Accuracy for class: cat   is 30.8 %
Accuracy for class: deer  is 43.7 %
Accuracy for class: dog   is 41.9 %
Accuracy for class: frog  is 57.6 %
Accuracy for class: horse is 60.2 %
Accuracy for class: ship  is 67.4 %
Accuracy for class: truck is 61.9 %
```