

Part 1: Fully connected neural networks (25 Points)

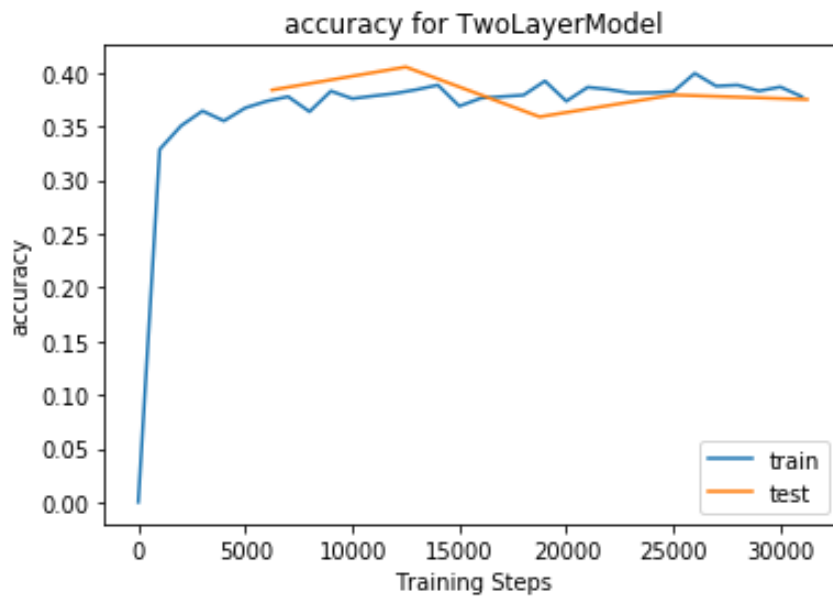
Test (on validation set) accuracy (5 Points): 0.375

Test loss (5 Points): 1.737857

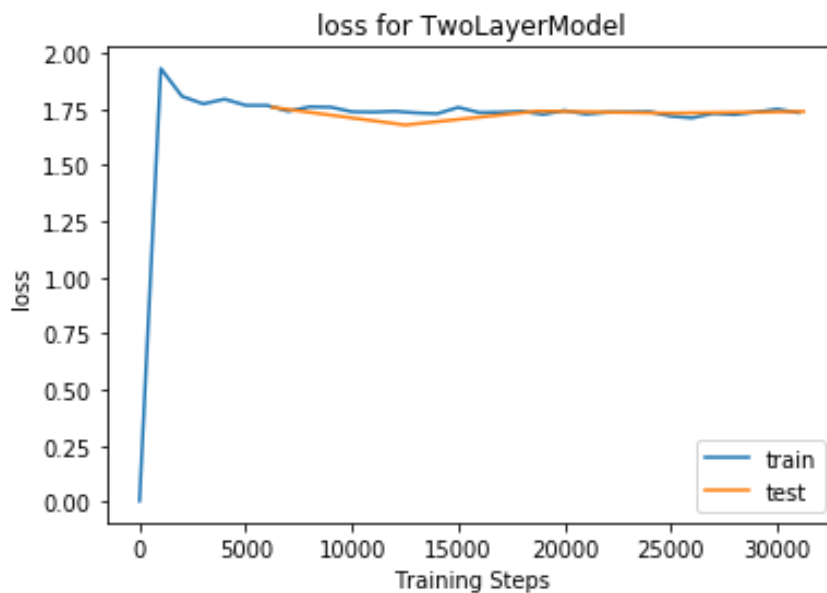
Training time (5 Points): 106.2 second in total

Plots:

- Plot a graph of accuracy on validation set vs training steps (5 Points)



- Plot a graph of loss on validation set vs training steps (5 Points)



Part 2: Convolution Network (Basic) (35 Points)

```
Output dimension after 1st conv layer:torch.Size([16, 32, 32])
Output dimension after 1st max pooling:torch.Size([16, 16, 16])
Output dimension after 2nd conv layer:torch.Size([16, 16, 16])
Output dimension after flatten layer:torch.Size([4096])
Output dimension after 1st fully connected layer:torch.Size([64])
Output dimension after 2nd fully connected layer:torch.Size([10])
```

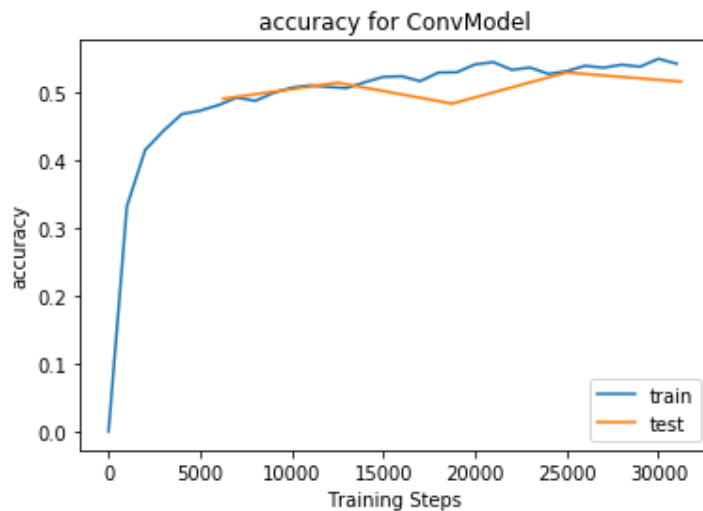
Test (on validation set) Accuracy (5 Points): 0.5148

Test loss (5 Points): 1.351214

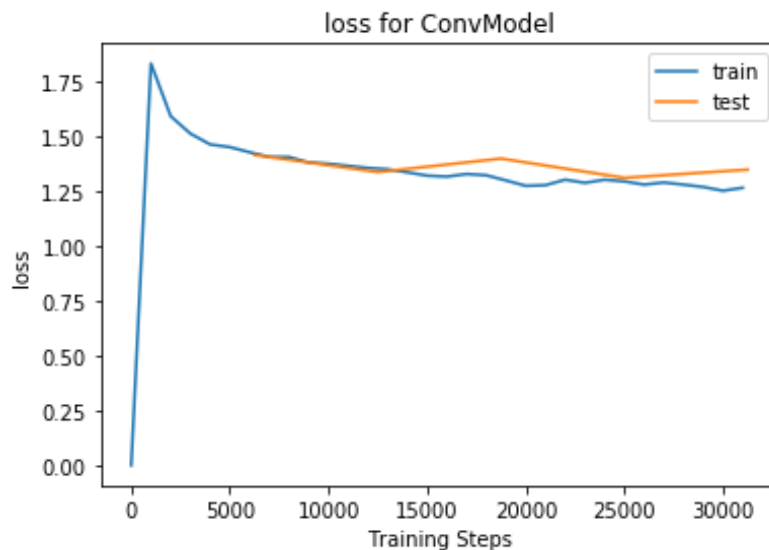
Training time (5 Points): 204.1 second in total

Plots:

- Plot a graph of accuracy on validation set vs training steps (5 Points)



- Plot a graph of loss on validation set vs training steps (5 Points)



Part 3: Convolution Network

1. Change out channel to 64(increase accuracy from experience)
2. Add additional convolution layer
3. change learning rate from 0,001 to 0.0005 and weight decay from 0.01 to 0.001(original learning rate and weight decay might be to high)
4. Increase epoch to 30 (5 was too little, tried 50 as well, but no obvious improvement after 30)
5. Try increase max pool window to 4, since image was not big anyway, pooling from 4x4 delete too much information and causes accuracy to drop.

Test (on validation set) Accuracy (5 Points): 0.738709

Test loss (5 Points): 1.029158

Training time (5 Points): 497.4 second in total

