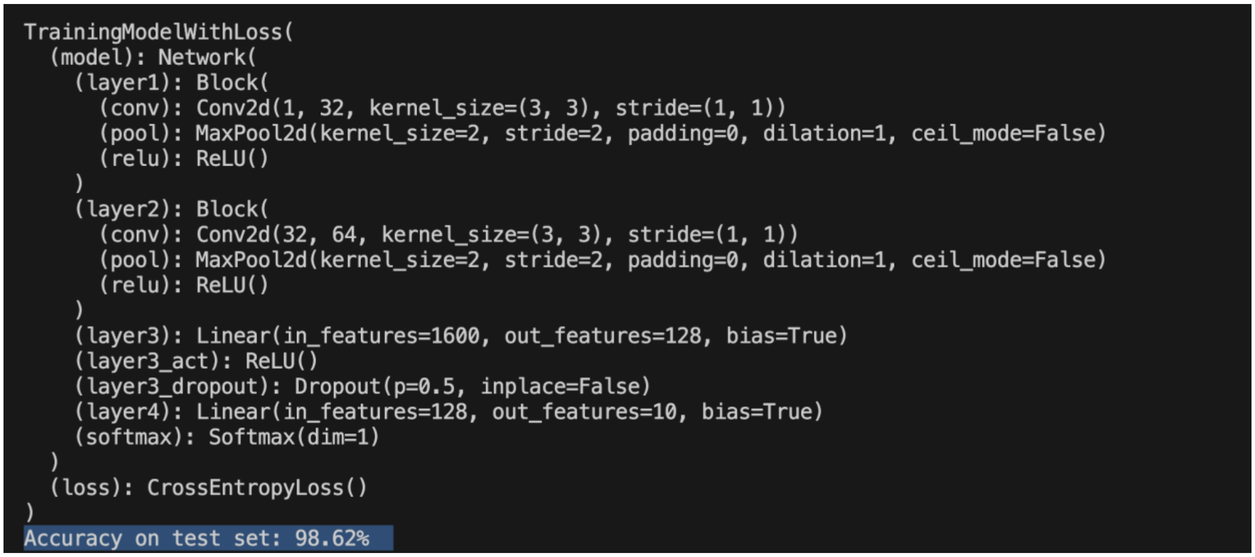
Graphcore

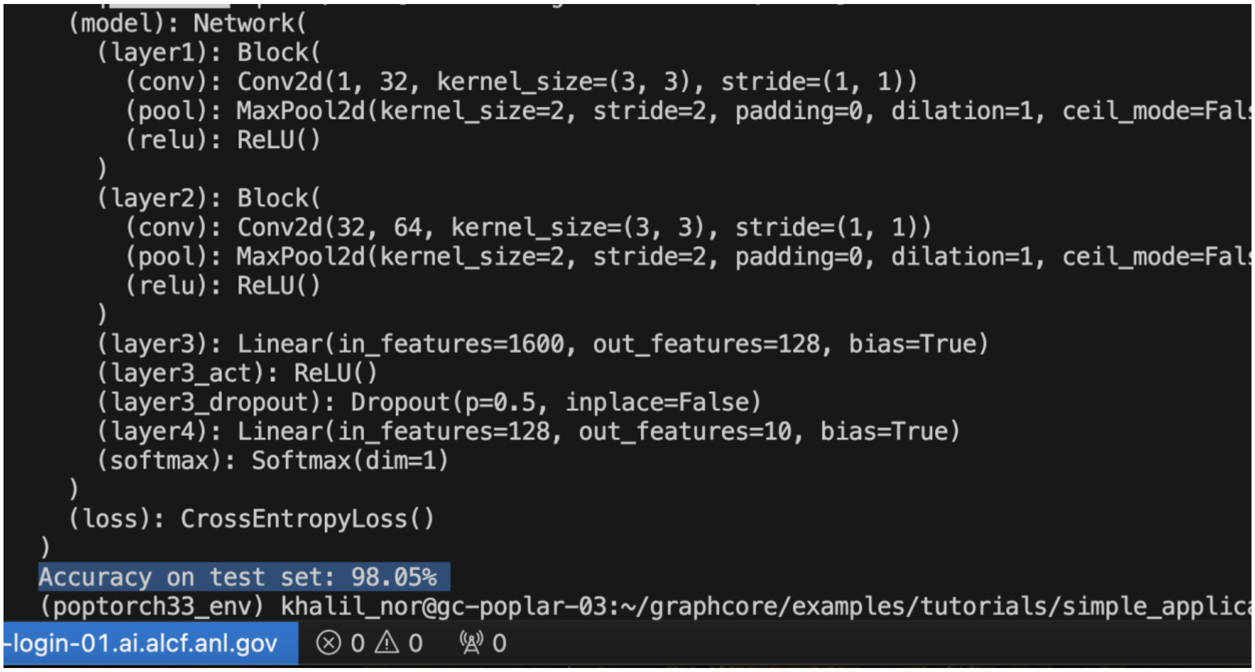
Initial results (running mnist_poptorch.py as is):



```
TrainingModelWithLoss(
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 98.62%
```

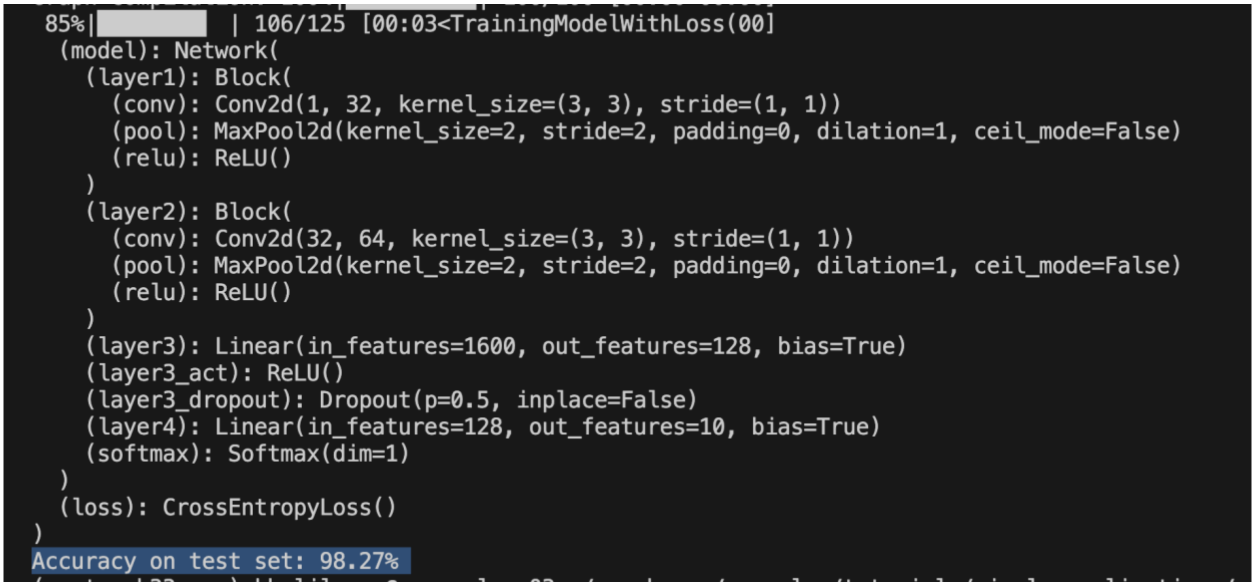
Changing batch size:

Batch_size = 24 (below)



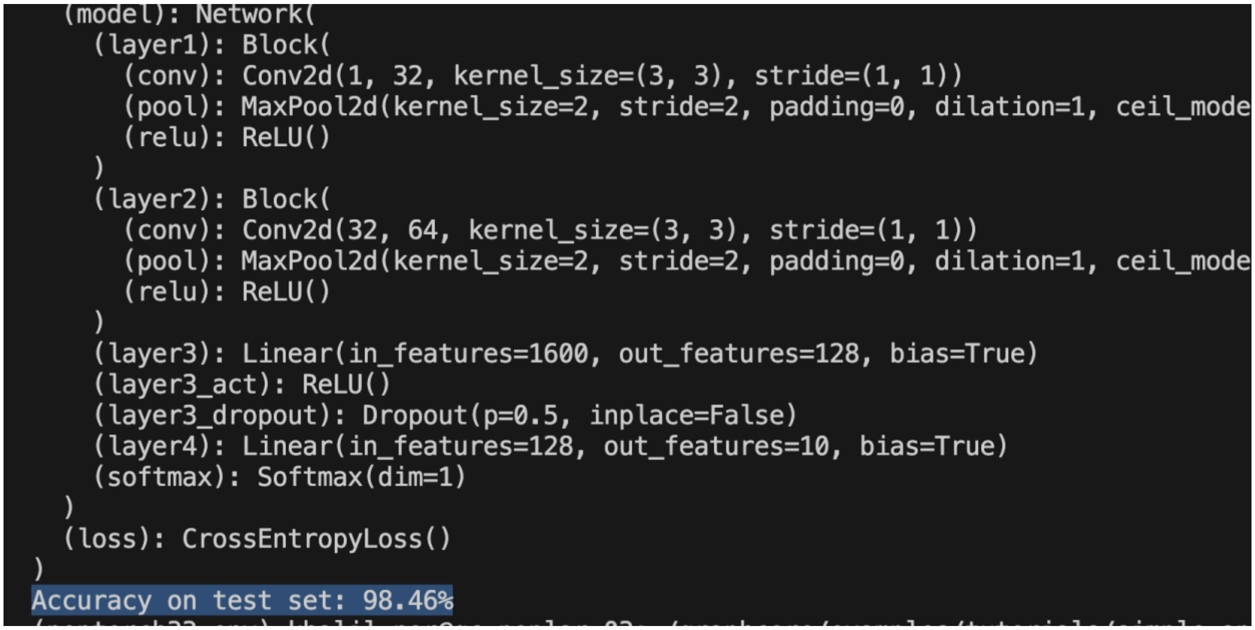
```
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=Fal
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=Fal
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 98.05%
(poptorch33_env) khalil_nor@gc-poplar-03:~/graphcore/examples/tutorials/simple_applica
```

Batch_size = 56 (below)



```
 85%|████████  | 106/125 [00:03<TrainingModelWithLoss(00]
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 98.27%
```

Batch_size = 128

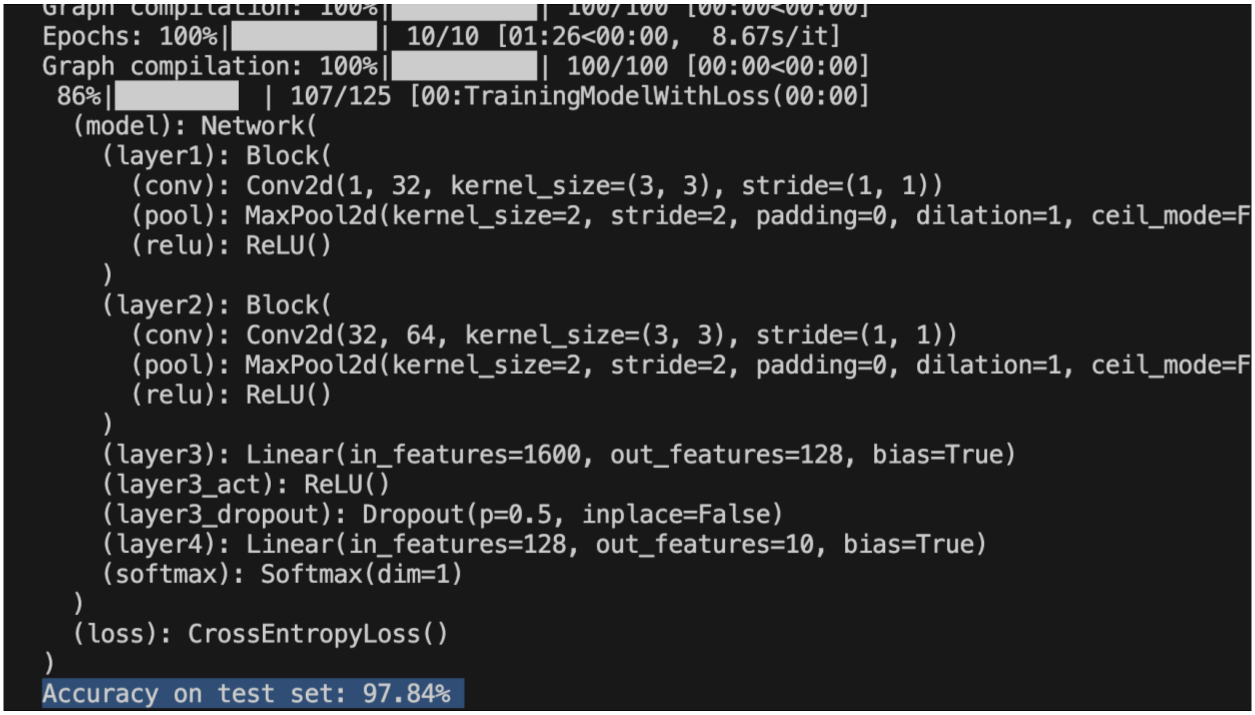


```
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 98.46%
```

When set to its initial batch size of 8, the model exhibited the highest test accuracy (98.62%). Generally, except for the initial case, increasing the batch size from 24 to 56 to 128 lead to small increases in test accuracy, although these were very minimal.

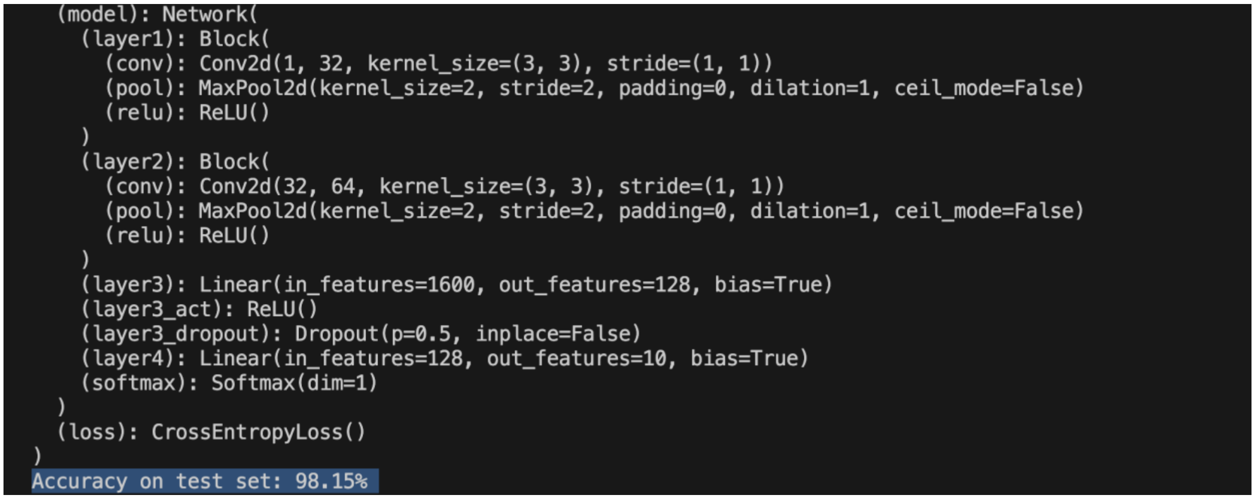
Changing learning rate:

Lr = 0.06



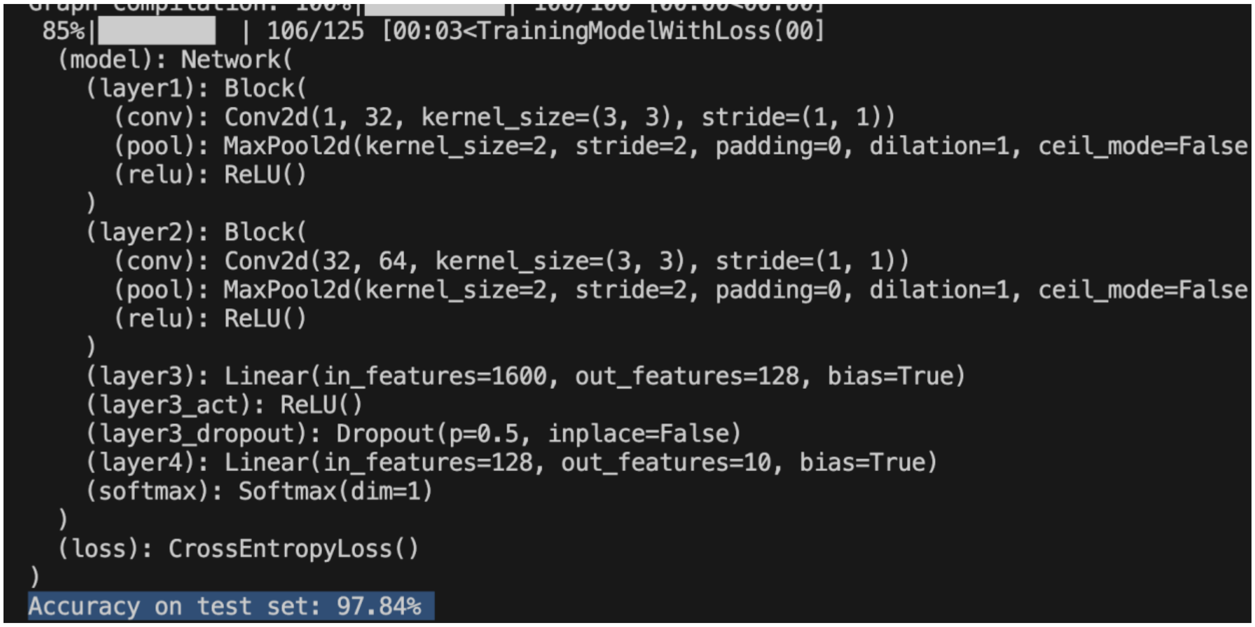
```
Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
Epochs: 100%|██████████| 10/10 [01:26<00:00,  8.67s/it]
Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
 86%|████████  | 107/125 [00:TrainingModelWithLoss(00:00]
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=F
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=F
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 97.84%
```

Lr = 0.12



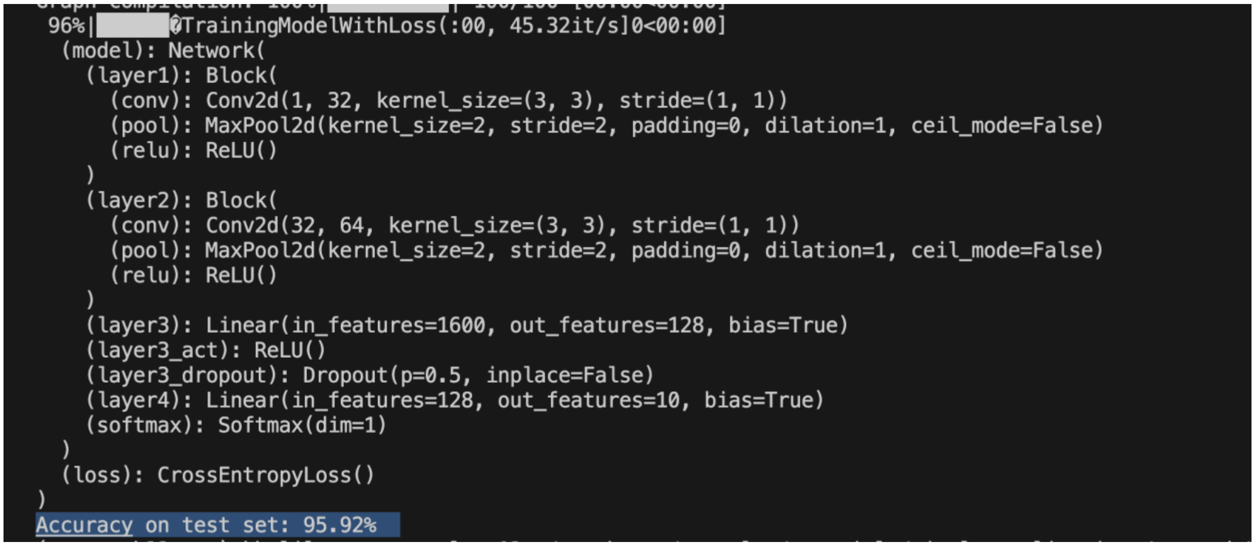
```
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 98.15%
```

Lr = 0.25



```
 Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
  85%|████████  | 106/125 [00:03<TrainingModelWithLoss(00]
   (model): Network(
     (layer1): Block(
       (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
       (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False
       (relu): ReLU()
     )
     (layer2): Block(
       (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
       (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False
       (relu): ReLU()
     )
     (layer3): Linear(in_features=1600, out_features=128, bias=True)
     (layer3_act): ReLU()
     (layer3_dropout): Dropout(p=0.5, inplace=False)
     (layer4): Linear(in_features=128, out_features=10, bias=True)
     (softmax): Softmax(dim=1)
   )
   (loss): CrossEntropyLoss()
 )
 Accuracy on test set: 97.84%
```

Lr = 0.35

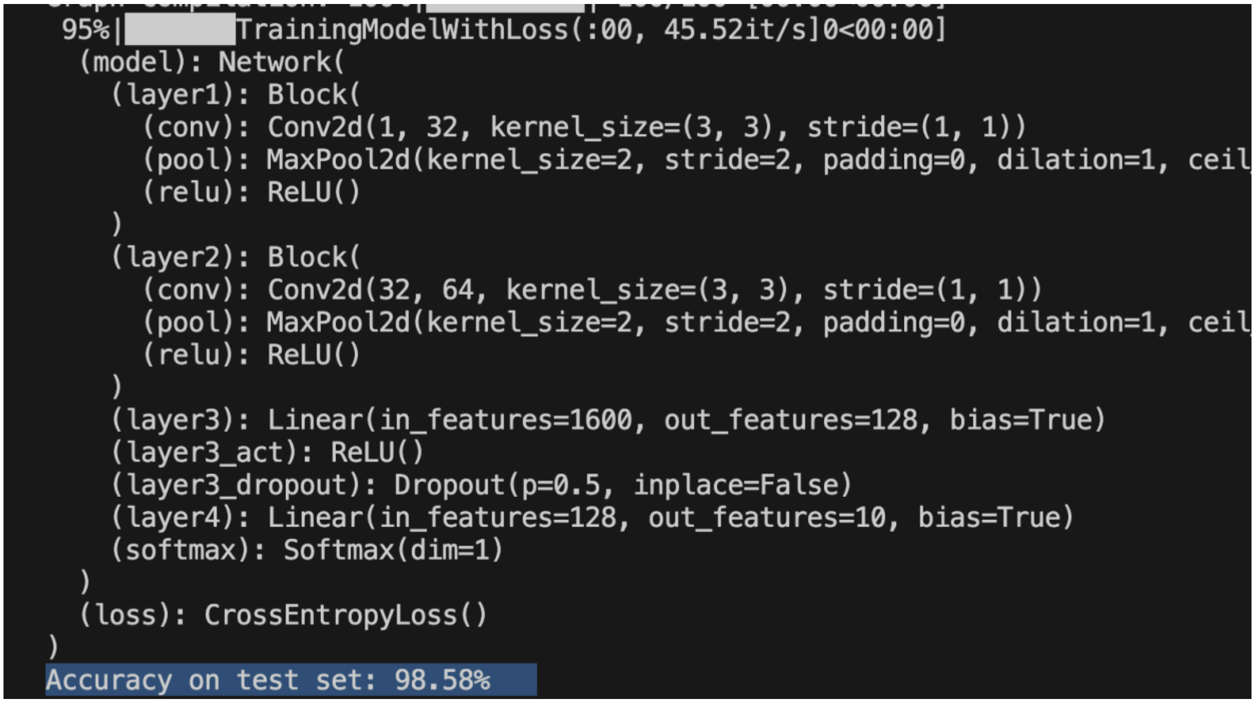


```
  Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
   96%|██████�TrainingModelWithLoss(:00, 45.32it/s]0<00:00]
    (model): Network(
      (layer1): Block(
        (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
        (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (relu): ReLU()
      )
      (layer2): Block(
        (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
        (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
        (relu): ReLU()
      )
      (layer3): Linear(in_features=1600, out_features=128, bias=True)
      (layer3_act): ReLU()
      (layer3_dropout): Dropout(p=0.5, inplace=False)
      (layer4): Linear(in_features=128, out_features=10, bias=True)
      (softmax): Softmax(dim=1)
    )
    (loss): CrossEntropyLoss()
  )
  Accuracy on test set: 95.92%
```

Greatly increasing the learning rate significantly decreases the accuracy on the test set.

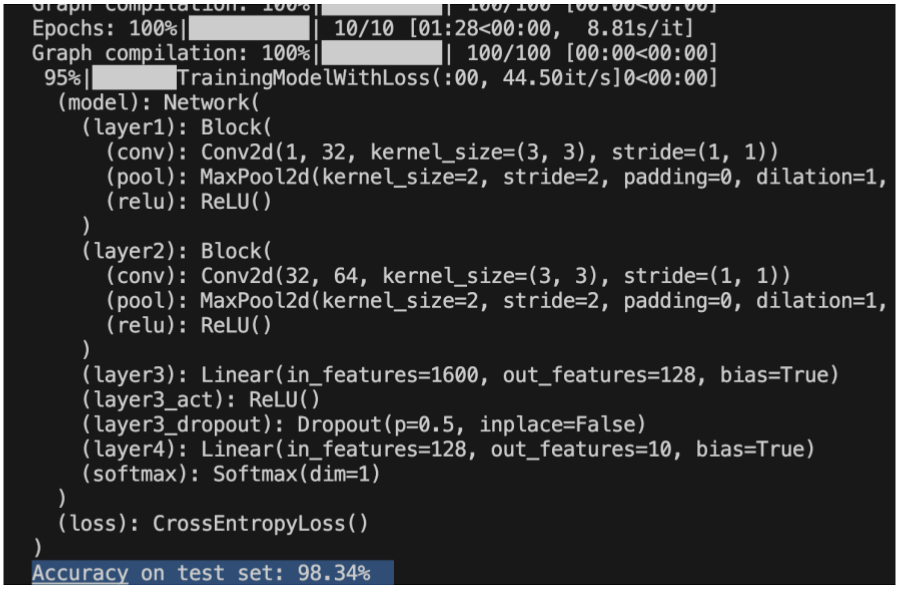
Changing number of epochs:

Epochs = 4



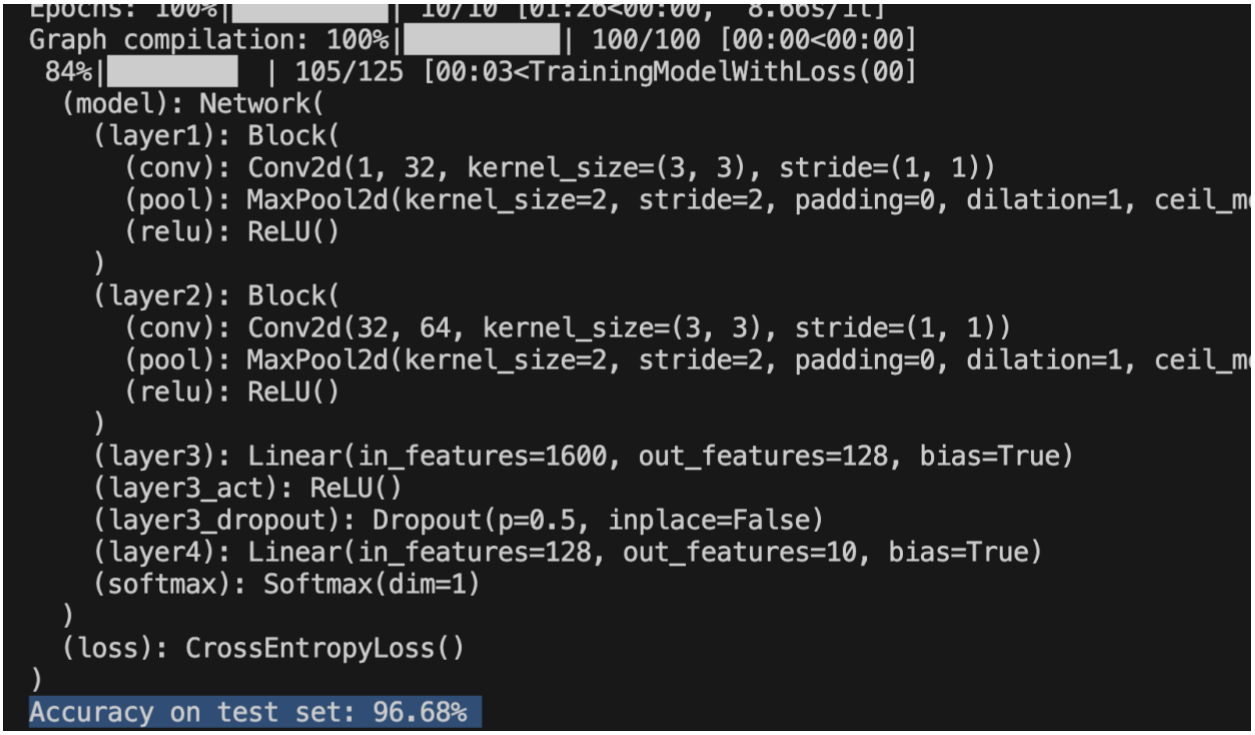
```
 95%|███████ TrainingModelWithLoss(:00, 45.52it/s]0<00:00]
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 98.58%
```

Epochs = 7



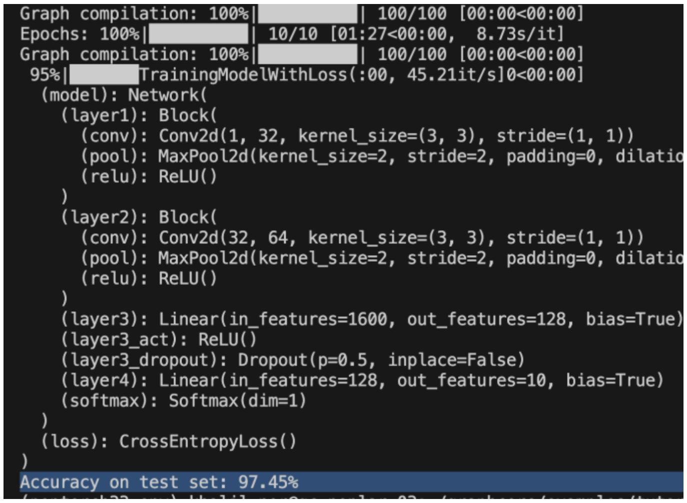
```
Epochs: 100%|██████████| 10/10 [01:28<00:00,  8.81s/it]
Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
 95%|███████ TrainingModelWithLoss(:00, 44.50it/s]0<00:00]
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 98.34%
```

Epochs = 12



```
Epochs: 100%|██████████| 10/10 [01:26<00:00,  8.66s/it]
Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
 84%|████████   | 105/125 [00:03<TrainingModelWithLoss(00]
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_m
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_m
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 96.68%
```

Epochs = 20



```
Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
Epochs: 100%|██████████| 10/10 [01:27<00:00,  8.73s/it]
Graph compilation: 100%|██████████| 100/100 [00:00<00:00]
 95%|███████TrainingModelWithLoss(:00, 45.21it/s]0<00:00]
  (model): Network(
    (layer1): Block(
      (conv): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilatio
      (relu): ReLU()
    )
    (layer2): Block(
      (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
      (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilatio
      (relu): ReLU()
    )
    (layer3): Linear(in_features=1600, out_features=128, bias=True)
    (layer3_act): ReLU()
    (layer3_dropout): Dropout(p=0.5, inplace=False)
    (layer4): Linear(in_features=128, out_features=10, bias=True)
    (softmax): Softmax(dim=1)
  )
  (loss): CrossEntropyLoss()
)
Accuracy on test set: 97.45%
```

Increasing to be > 10 resulted in a decreased test accuracy, while keeping the epochs between 4 – 10 resulted in a test accuracy that was basically the same.