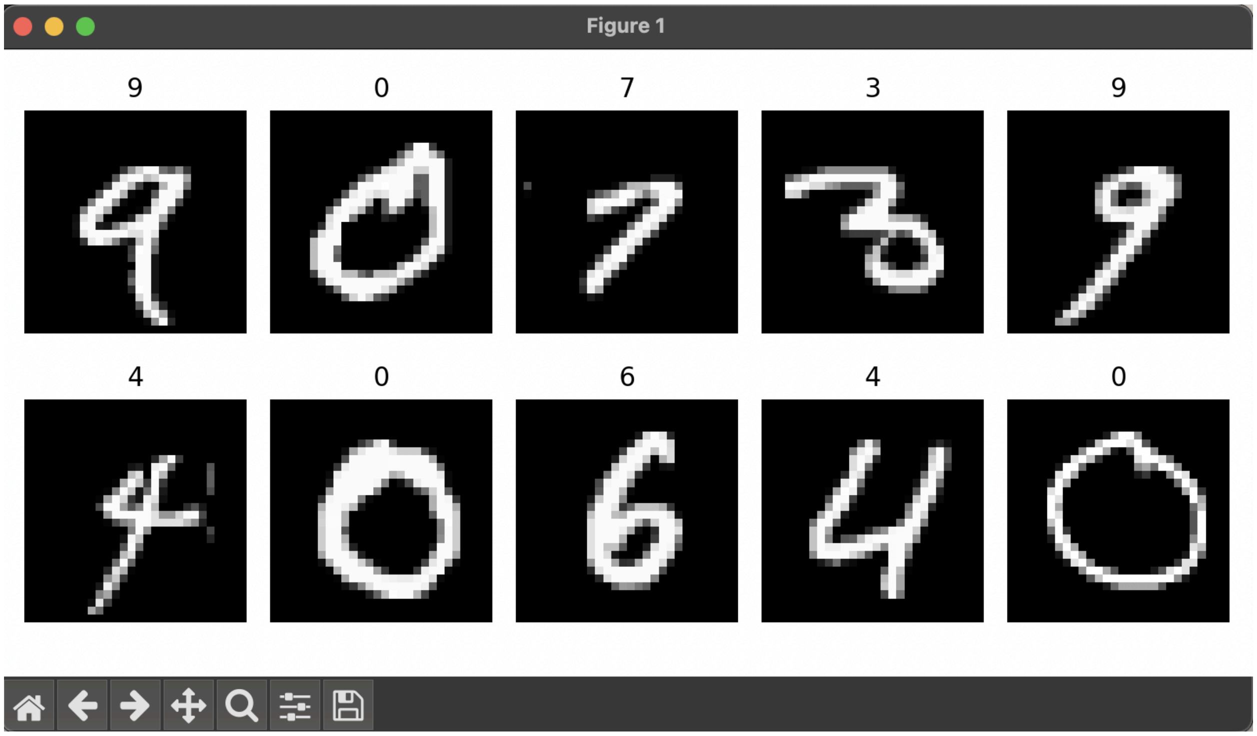
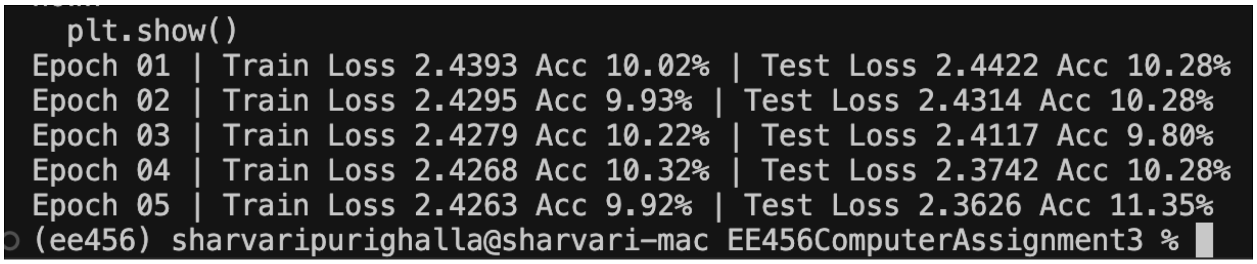
Part A: MLP

Task 1



Task 2



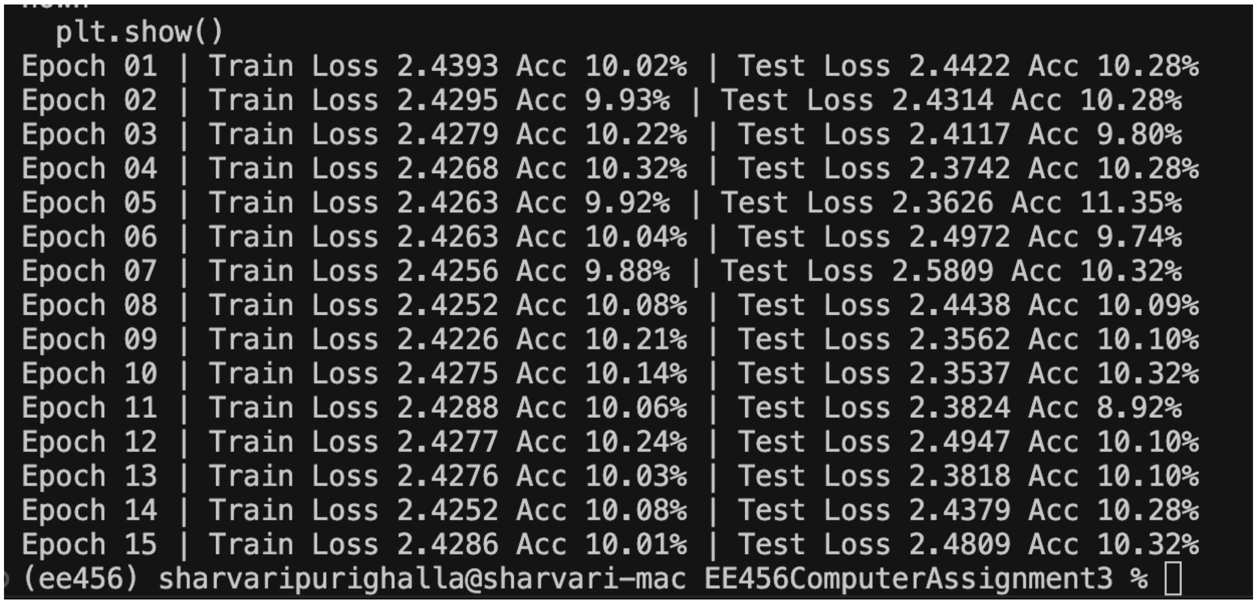
```
  plt.show()
Epoch 01 | Train Loss 2.4393 Acc 10.02% | Test Loss 2.4422 Acc 10.28%
Epoch 02 | Train Loss 2.4295 Acc 9.93% | Test Loss 2.4314 Acc 10.28%
Epoch 03 | Train Loss 2.4279 Acc 10.22% | Test Loss 2.4117 Acc 9.80%
Epoch 04 | Train Loss 2.4268 Acc 10.32% | Test Loss 2.3742 Acc 10.28%
Epoch 05 | Train Loss 2.4263 Acc 9.92% | Test Loss 2.3626 Acc 11.35%
(ee456) sharvaripurighalla@sharvari-mac EE456ComputerAssignment3 %
```

The training and test accuracies remain around 10%, which is roughly the same as random guessing among ten digit classes (0-9). Loss values stay near 2.43 → 2.36, showing no meaningful learning trend.
This happens because:

- The MLP is extremely small (784 → 8 → 4 → 10) with very few neurons to capture complex digit patterns
- The learning rate (0.5) is too high, causing unstable or ineffective gradient updates.
- The only 5 epochs of training provide very limited optimization steps

Thus, this result is **not good**. The network is effectively guessing and has not learned to recognize digits. It's a baseline that confirms the model and training loop run correctly but requires larger capacity, lower learning rate, or more epochs to perform well.

Task 3

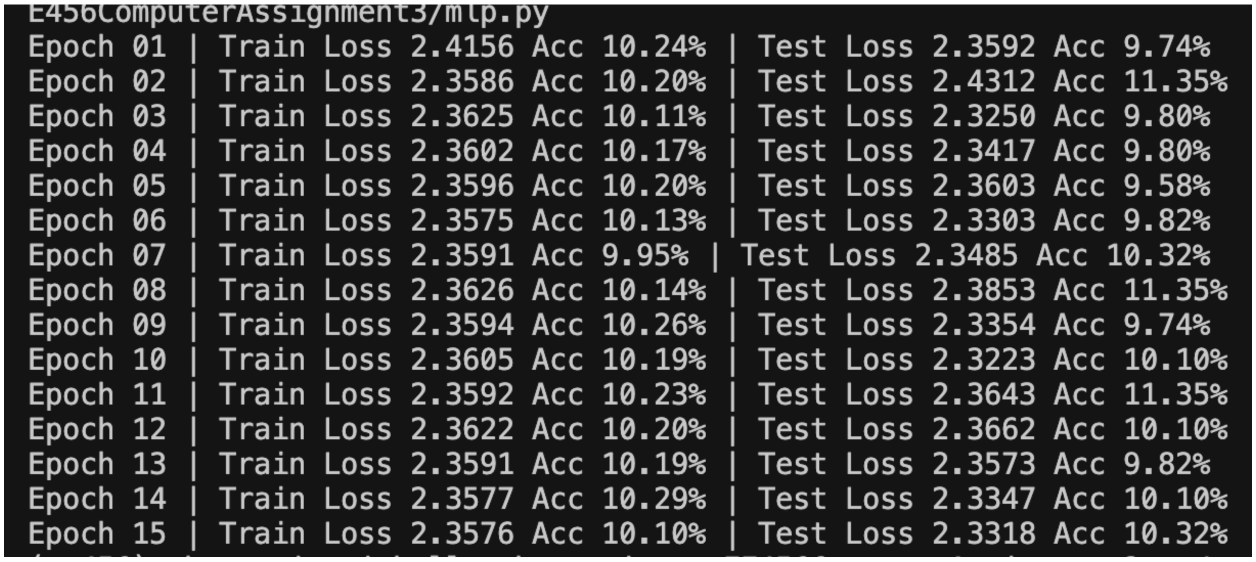


```
  plt.show()
Epoch 01 | Train Loss 2.4393 Acc 10.02% | Test Loss 2.4422 Acc 10.28%
Epoch 02 | Train Loss 2.4295 Acc 9.93% | Test Loss 2.4314 Acc 10.28%
Epoch 03 | Train Loss 2.4279 Acc 10.22% | Test Loss 2.4117 Acc 9.80%
Epoch 04 | Train Loss 2.4268 Acc 10.32% | Test Loss 2.3742 Acc 10.28%
Epoch 05 | Train Loss 2.4263 Acc 9.92% | Test Loss 2.3626 Acc 11.35%
Epoch 06 | Train Loss 2.4263 Acc 10.04% | Test Loss 2.4972 Acc 9.74%
Epoch 07 | Train Loss 2.4256 Acc 9.88% | Test Loss 2.5809 Acc 10.32%
Epoch 08 | Train Loss 2.4252 Acc 10.08% | Test Loss 2.4438 Acc 10.09%
Epoch 09 | Train Loss 2.4226 Acc 10.21% | Test Loss 2.3562 Acc 10.10%
Epoch 10 | Train Loss 2.4275 Acc 10.14% | Test Loss 2.3537 Acc 10.32%
Epoch 11 | Train Loss 2.4288 Acc 10.06% | Test Loss 2.3824 Acc 8.92%
Epoch 12 | Train Loss 2.4277 Acc 10.24% | Test Loss 2.4947 Acc 10.10%
Epoch 13 | Train Loss 2.4276 Acc 10.03% | Test Loss 2.3818 Acc 10.10%
Epoch 14 | Train Loss 2.4252 Acc 10.08% | Test Loss 2.4379 Acc 10.28%
Epoch 15 | Train Loss 2.4286 Acc 10.01% | Test Loss 2.4809 Acc 10.32%
(ee456) sharvaripurighalla@sharvari-mac EE456ComputerAssignment3 %
```

Even after 15 epochs, both training and testing accuracy remain around 10%, which still reflects random guessing among the 10 digit classes. The loss value does not meaningfully decrease, suggesting the model is not learning any useful features. The network is still too shallow, providing insufficient representational capacity to capture MNIST digit patterns. The learning rate is also too high, and simply increasing the number of epochs does not help when the model capacity of learning rate are the true bottlenecks.

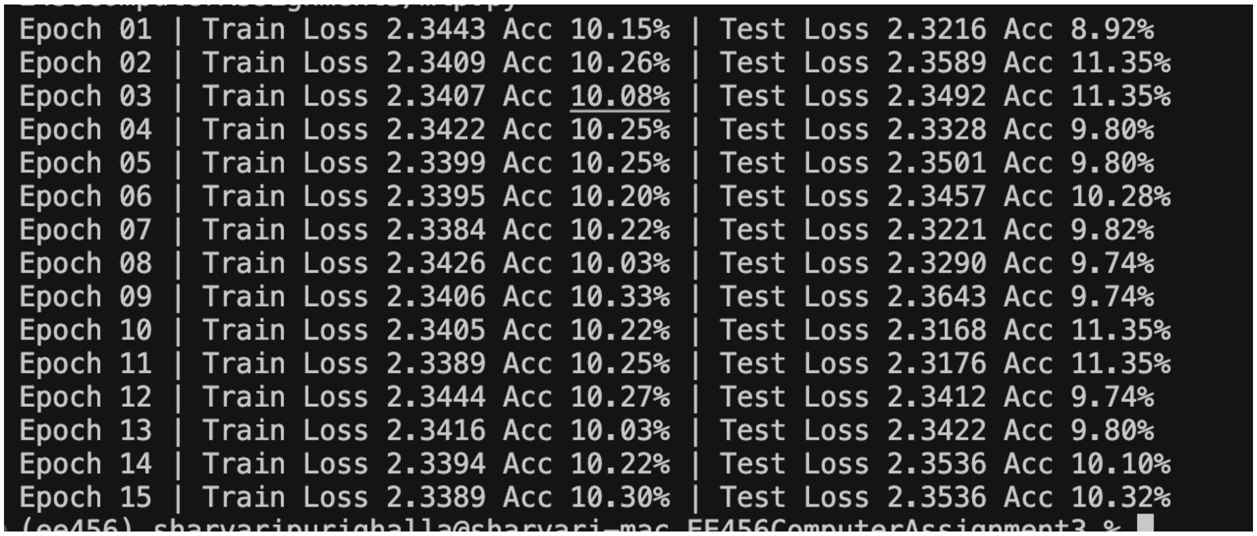
Task 4

→ Batch size = 32:



```
E456ComputerAssignment3/mlp.py
Epoch 01 | Train Loss 2.4156 Acc 10.24% | Test Loss 2.3592 Acc 9.74%
Epoch 02 | Train Loss 2.3586 Acc 10.20% | Test Loss 2.4312 Acc 11.35%
Epoch 03 | Train Loss 2.3625 Acc 10.11% | Test Loss 2.3250 Acc 9.80%
Epoch 04 | Train Loss 2.3602 Acc 10.17% | Test Loss 2.3417 Acc 9.80%
Epoch 05 | Train Loss 2.3596 Acc 10.20% | Test Loss 2.3603 Acc 9.58%
Epoch 06 | Train Loss 2.3575 Acc 10.13% | Test Loss 2.3303 Acc 9.82%
Epoch 07 | Train Loss 2.3591 Acc 9.95% | Test Loss 2.3485 Acc 10.32%
Epoch 08 | Train Loss 2.3626 Acc 10.14% | Test Loss 2.3853 Acc 11.35%
Epoch 09 | Train Loss 2.3594 Acc 10.26% | Test Loss 2.3354 Acc 9.74%
Epoch 10 | Train Loss 2.3605 Acc 10.19% | Test Loss 2.3223 Acc 10.10%
Epoch 11 | Train Loss 2.3592 Acc 10.23% | Test Loss 2.3643 Acc 11.35%
Epoch 12 | Train Loss 2.3622 Acc 10.20% | Test Loss 2.3662 Acc 10.10%
Epoch 13 | Train Loss 2.3591 Acc 10.19% | Test Loss 2.3573 Acc 9.82%
Epoch 14 | Train Loss 2.3577 Acc 10.29% | Test Loss 2.3347 Acc 10.10%
Epoch 15 | Train Loss 2.3576 Acc 10.10% | Test Loss 2.3318 Acc 10.32%
```

→ Batch size = 64:

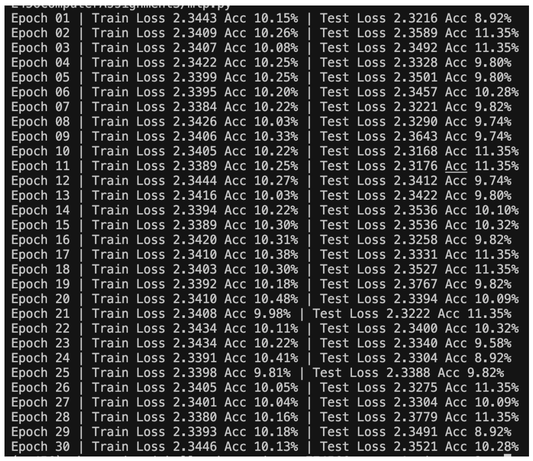


```
Epoch 01 | Train Loss 2.3443 Acc 10.15% | Test Loss 2.3216 Acc 8.92%
Epoch 02 | Train Loss 2.3409 Acc 10.26% | Test Loss 2.3589 Acc 11.35%
Epoch 03 | Train Loss 2.3407 Acc 10.08% | Test Loss 2.3492 Acc 11.35%
Epoch 04 | Train Loss 2.3422 Acc 10.25% | Test Loss 2.3328 Acc 9.80%
Epoch 05 | Train Loss 2.3399 Acc 10.25% | Test Loss 2.3501 Acc 9.80%
Epoch 06 | Train Loss 2.3395 Acc 10.20% | Test Loss 2.3457 Acc 10.28%
Epoch 07 | Train Loss 2.3384 Acc 10.22% | Test Loss 2.3221 Acc 9.82%
Epoch 08 | Train Loss 2.3426 Acc 10.03% | Test Loss 2.3290 Acc 9.74%
Epoch 09 | Train Loss 2.3406 Acc 10.33% | Test Loss 2.3643 Acc 9.74%
Epoch 10 | Train Loss 2.3405 Acc 10.22% | Test Loss 2.3168 Acc 11.35%
Epoch 11 | Train Loss 2.3389 Acc 10.25% | Test Loss 2.3176 Acc 11.35%
Epoch 12 | Train Loss 2.3444 Acc 10.27% | Test Loss 2.3412 Acc 9.74%
Epoch 13 | Train Loss 2.3416 Acc 10.03% | Test Loss 2.3422 Acc 9.80%
Epoch 14 | Train Loss 2.3394 Acc 10.22% | Test Loss 2.3536 Acc 10.10%
Epoch 15 | Train Loss 2.3389 Acc 10.30% | Test Loss 2.3536 Acc 10.32%
(ee456) sharvaripurighalla@sharvari-mac EE456ComputerAssignment3 %
```

Accuracy: both runs give ~10% accuracy, which is again near random chance, meaning the model still isn't learning meaningful patterns.
Speed: with batch = 64, each epoch completed slightly faster because the model performed fewer weight-update steps per epoch (fewer batches total)
Loss trend: loss values stayed nearly constant, confirming the learning rate is still too high or the model is too simple.

Thus, increasing the batch size did not improve accuracy but reduced raining time per epoch, which matches my theoretical expectations.

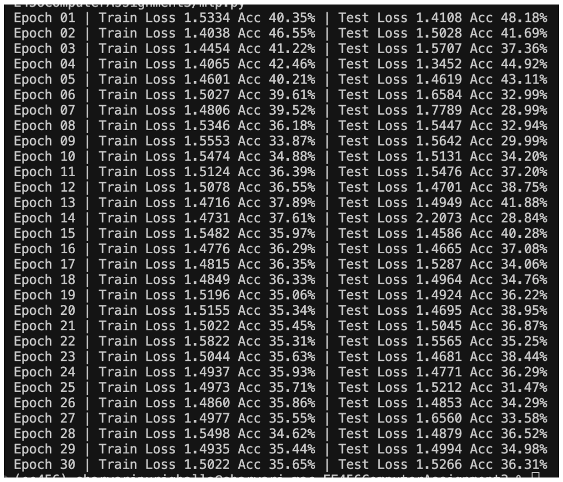
Task 5
→ 0.5 Learning Rate:



```
Epoch 01 | Train Loss 2.3443 Acc 10.15% | Test Loss 2.3216 Acc 8.92%
Epoch 02 | Train Loss 2.3409 Acc 10.26% | Test Loss 2.3589 Acc 11.35%
Epoch 03 | Train Loss 2.3407 Acc 10.08% | Test Loss 2.3492 Acc 11.35%
Epoch 04 | Train Loss 2.3422 Acc 10.25% | Test Loss 2.3328 Acc 9.80%
Epoch 05 | Train Loss 2.3399 Acc 10.25% | Test Loss 2.3501 Acc 9.80%
Epoch 06 | Train Loss 2.3395 Acc 10.20% | Test Loss 2.3457 Acc 10.28%
Epoch 07 | Train Loss 2.3384 Acc 10.22% | Test Loss 2.3221 Acc 9.82%
Epoch 08 | Train Loss 2.3426 Acc 10.03% | Test Loss 2.3290 Acc 9.74%
Epoch 09 | Train Loss 2.3406 Acc 10.33% | Test Loss 2.3643 Acc 9.74%
Epoch 10 | Train Loss 2.3405 Acc 10.22% | Test Loss 2.3168 Acc 11.35%
Epoch 11 | Train Loss 2.3389 Acc 10.25% | Test Loss 2.3176 Acc 11.35%
Epoch 12 | Train Loss 2.3444 Acc 10.27% | Test Loss 2.3412 Acc 9.74%
Epoch 13 | Train Loss 2.3416 Acc 10.03% | Test Loss 2.3422 Acc 9.80%
Epoch 14 | Train Loss 2.3394 Acc 10.22% | Test Loss 2.3536 Acc 10.10%
Epoch 15 | Train Loss 2.3389 Acc 10.30% | Test Loss 2.3536 Acc 10.32%
Epoch 16 | Train Loss 2.3420 Acc 10.31% | Test Loss 2.3258 Acc 9.82%
Epoch 17 | Train Loss 2.3410 Acc 10.38% | Test Loss 2.3331 Acc 11.35%
Epoch 18 | Train Loss 2.3403 Acc 10.30% | Test Loss 2.3527 Acc 11.35%
Epoch 19 | Train Loss 2.3392 Acc 10.18% | Test Loss 2.3767 Acc 9.82%
Epoch 20 | Train Loss 2.3410 Acc 10.48% | Test Loss 2.3394 Acc 10.09%
Epoch 21 | Train Loss 2.3408 Acc 9.98% | Test Loss 2.3222 Acc 11.35%
Epoch 22 | Train Loss 2.3434 Acc 10.11% | Test Loss 2.3400 Acc 10.32%
Epoch 23 | Train Loss 2.3434 Acc 10.22% | Test Loss 2.3340 Acc 9.58%
Epoch 24 | Train Loss 2.3391 Acc 10.41% | Test Loss 2.3304 Acc 8.92%
Epoch 25 | Train Loss 2.3398 Acc 9.81% | Test Loss 2.3388 Acc 9.82%
Epoch 26 | Train Loss 2.3405 Acc 10.05% | Test Loss 2.3275 Acc 11.35%
Epoch 27 | Train Loss 2.3401 Acc 10.04% | Test Loss 2.3304 Acc 10.09%
Epoch 28 | Train Loss 2.3380 Acc 10.16% | Test Loss 2.3779 Acc 11.35%
Epoch 29 | Train Loss 2.3393 Acc 10.18% | Test Loss 2.3491 Acc 8.92%
Epoch 30 | Train Loss 2.3446 Acc 10.13% | Test Loss 2.3521 Acc 10.28%
```

- Training accuracy ~ 10%, Test accuracy ~ 20%
- Behavior: no learning. Loss ~ 2.34 constant
- Reason: the step size is too large; gradients overshoot the optimum, so weights never converge

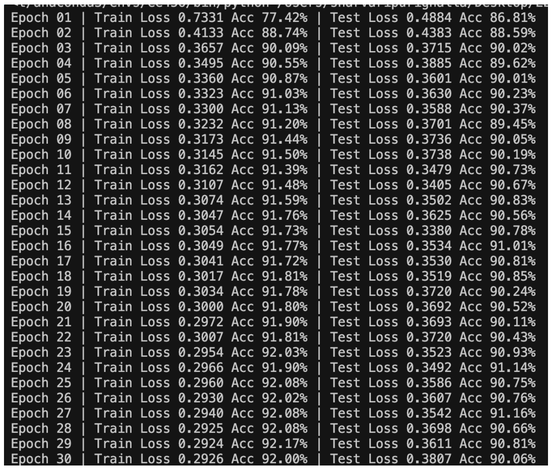
→ 0.1 Learning Rate:



```
Epoch 01 | Train Loss 1.5334 Acc 40.35% | Test Loss 1.4108 Acc 48.18%
Epoch 02 | Train Loss 1.4038 Acc 46.55% | Test Loss 1.5028 Acc 41.69%
Epoch 03 | Train Loss 1.4454 Acc 41.22% | Test Loss 1.5707 Acc 37.36%
Epoch 04 | Train Loss 1.4065 Acc 42.46% | Test Loss 1.3452 Acc 44.92%
Epoch 05 | Train Loss 1.4601 Acc 40.21% | Test Loss 1.4619 Acc 43.11%
Epoch 06 | Train Loss 1.5027 Acc 39.61% | Test Loss 1.6584 Acc 32.99%
Epoch 07 | Train Loss 1.4806 Acc 39.52% | Test Loss 1.7789 Acc 28.99%
Epoch 08 | Train Loss 1.5346 Acc 36.18% | Test Loss 1.5447 Acc 32.94%
Epoch 09 | Train Loss 1.5553 Acc 33.87% | Test Loss 1.5642 Acc 29.99%
Epoch 10 | Train Loss 1.5474 Acc 34.88% | Test Loss 1.5131 Acc 34.20%
Epoch 11 | Train Loss 1.5124 Acc 36.39% | Test Loss 1.5476 Acc 37.20%
Epoch 12 | Train Loss 1.5078 Acc 36.55% | Test Loss 1.4701 Acc 38.75%
Epoch 13 | Train Loss 1.4716 Acc 37.89% | Test Loss 1.4949 Acc 41.88%
Epoch 14 | Train Loss 1.4731 Acc 37.61% | Test Loss 2.2073 Acc 28.84%
Epoch 15 | Train Loss 1.5482 Acc 35.97% | Test Loss 1.4586 Acc 40.28%
Epoch 16 | Train Loss 1.4776 Acc 36.29% | Test Loss 1.4665 Acc 37.08%
Epoch 17 | Train Loss 1.4815 Acc 36.35% | Test Loss 1.5287 Acc 34.06%
Epoch 18 | Train Loss 1.4849 Acc 36.33% | Test Loss 1.4964 Acc 34.76%
Epoch 19 | Train Loss 1.5196 Acc 35.06% | Test Loss 1.4924 Acc 36.22%
Epoch 20 | Train Loss 1.5155 Acc 35.34% | Test Loss 1.4695 Acc 38.95%
Epoch 21 | Train Loss 1.5022 Acc 35.45% | Test Loss 1.5045 Acc 36.87%
Epoch 22 | Train Loss 1.5822 Acc 35.31% | Test Loss 1.5565 Acc 35.25%
Epoch 23 | Train Loss 1.5044 Acc 35.63% | Test Loss 1.4681 Acc 38.44%
Epoch 24 | Train Loss 1.4937 Acc 35.93% | Test Loss 1.4771 Acc 36.29%
Epoch 25 | Train Loss 1.4973 Acc 35.71% | Test Loss 1.5212 Acc 31.47%
Epoch 26 | Train Loss 1.4860 Acc 35.86% | Test Loss 1.4853 Acc 34.29%
Epoch 27 | Train Loss 1.4977 Acc 35.55% | Test Loss 1.6560 Acc 33.58%
Epoch 28 | Train Loss 1.5498 Acc 34.62% | Test Loss 1.4879 Acc 36.52%
Epoch 29 | Train Loss 1.4935 Acc 35.44% | Test Loss 1.4994 Acc 34.98%
Epoch 30 | Train Loss 1.5022 Acc 35.65% | Test Loss 1.5266 Acc 36.31%
```

- Training accuracy ~ 35 - 46%, Test Accuracy ~ 30 - 48%
- Behavior: model started to learn but accuracy fluctuated and plateaued early.
- Reason: still relatively high LR → unstable updates and oscillation around the optimum

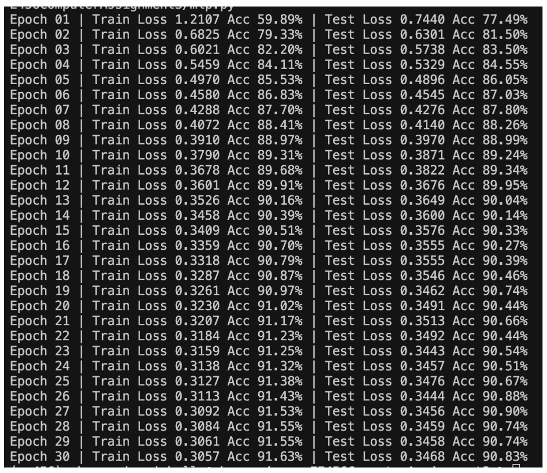
→ 0.01 Learning Rate:



```
Epoch 01 | Train Loss 0.7331 Acc 77.42% | Test Loss 0.4884 Acc 86.81%
Epoch 02 | Train Loss 0.4133 Acc 88.74% | Test Loss 0.4383 Acc 88.59%
Epoch 03 | Train Loss 0.3657 Acc 90.09% | Test Loss 0.3715 Acc 90.02%
Epoch 04 | Train Loss 0.3495 Acc 90.55% | Test Loss 0.3885 Acc 89.62%
Epoch 05 | Train Loss 0.3360 Acc 90.87% | Test Loss 0.3601 Acc 90.01%
Epoch 06 | Train Loss 0.3323 Acc 91.03% | Test Loss 0.3630 Acc 90.23%
Epoch 07 | Train Loss 0.3300 Acc 91.13% | Test Loss 0.3588 Acc 90.37%
Epoch 08 | Train Loss 0.3232 Acc 91.20% | Test Loss 0.3701 Acc 89.45%
Epoch 09 | Train Loss 0.3173 Acc 91.44% | Test Loss 0.3736 Acc 90.05%
Epoch 10 | Train Loss 0.3145 Acc 91.50% | Test Loss 0.3738 Acc 90.19%
Epoch 11 | Train Loss 0.3162 Acc 91.39% | Test Loss 0.3479 Acc 90.73%
Epoch 12 | Train Loss 0.3107 Acc 91.48% | Test Loss 0.3405 Acc 90.67%
Epoch 13 | Train Loss 0.3074 Acc 91.59% | Test Loss 0.3502 Acc 90.83%
Epoch 14 | Train Loss 0.3047 Acc 91.76% | Test Loss 0.3625 Acc 90.56%
Epoch 15 | Train Loss 0.3054 Acc 91.73% | Test Loss 0.3380 Acc 90.78%
Epoch 16 | Train Loss 0.3049 Acc 91.77% | Test Loss 0.3534 Acc 91.01%
Epoch 17 | Train Loss 0.3041 Acc 91.72% | Test Loss 0.3530 Acc 90.81%
Epoch 18 | Train Loss 0.3017 Acc 91.81% | Test Loss 0.3519 Acc 90.85%
Epoch 19 | Train Loss 0.3034 Acc 91.78% | Test Loss 0.3720 Acc 90.24%
Epoch 20 | Train Loss 0.3000 Acc 91.80% | Test Loss 0.3692 Acc 90.52%
Epoch 21 | Train Loss 0.2972 Acc 91.90% | Test Loss 0.3693 Acc 90.11%
Epoch 22 | Train Loss 0.3007 Acc 91.81% | Test Loss 0.3720 Acc 90.43%
Epoch 23 | Train Loss 0.2954 Acc 92.03% | Test Loss 0.3523 Acc 90.93%
Epoch 24 | Train Loss 0.2966 Acc 91.90% | Test Loss 0.3492 Acc 91.14%
Epoch 25 | Train Loss 0.2960 Acc 92.08% | Test Loss 0.3586 Acc 90.75%
Epoch 26 | Train Loss 0.2930 Acc 92.02% | Test Loss 0.3607 Acc 90.76%
Epoch 27 | Train Loss 0.2940 Acc 92.08% | Test Loss 0.3542 Acc 91.16%
Epoch 28 | Train Loss 0.2925 Acc 92.08% | Test Loss 0.3698 Acc 90.66%
Epoch 29 | Train Loss 0.2924 Acc 92.17% | Test Loss 0.3611 Acc 90.81%
Epoch 30 | Train Loss 0.2926 Acc 92.00% | Test Loss 0.3807 Acc 90.06%
```

- Training accuracy: went up from 77% to 92%, Test accuracy: increased to ~91%
- Behavior: stable convergence with smooth loss reduction from 0.73 → 0.29
- Reason: LR = 0.01 provided a balanced trade-off between speed and stability, allowing the model to learn digit patterns effectively.

→ 0.001 Learning Rate:



```
Epoch 01 | Train Loss 1.2107 Acc 59.89% | Test Loss 0.7440 Acc 77.49%
Epoch 02 | Train Loss 0.6825 Acc 79.33% | Test Loss 0.6301 Acc 81.50%
Epoch 03 | Train Loss 0.6021 Acc 82.20% | Test Loss 0.5738 Acc 83.50%
Epoch 04 | Train Loss 0.5459 Acc 84.11% | Test Loss 0.5329 Acc 84.55%
Epoch 05 | Train Loss 0.4970 Acc 85.53% | Test Loss 0.4896 Acc 86.05%
Epoch 06 | Train Loss 0.4580 Acc 86.83% | Test Loss 0.4545 Acc 87.03%
Epoch 07 | Train Loss 0.4288 Acc 87.70% | Test Loss 0.4276 Acc 87.80%
Epoch 08 | Train Loss 0.4072 Acc 88.41% | Test Loss 0.4140 Acc 88.26%
Epoch 09 | Train Loss 0.3910 Acc 88.97% | Test Loss 0.3970 Acc 88.99%
Epoch 10 | Train Loss 0.3790 Acc 89.31% | Test Loss 0.3871 Acc 89.24%
Epoch 11 | Train Loss 0.3678 Acc 89.68% | Test Loss 0.3822 Acc 89.34%
Epoch 12 | Train Loss 0.3601 Acc 89.91% | Test Loss 0.3676 Acc 89.95%
Epoch 13 | Train Loss 0.3526 Acc 90.16% | Test Loss 0.3649 Acc 90.04%
Epoch 14 | Train Loss 0.3458 Acc 90.39% | Test Loss 0.3600 Acc 90.14%
Epoch 15 | Train Loss 0.3409 Acc 90.51% | Test Loss 0.3576 Acc 90.33%
Epoch 16 | Train Loss 0.3359 Acc 90.70% | Test Loss 0.3555 Acc 90.27%
Epoch 17 | Train Loss 0.3318 Acc 90.79% | Test Loss 0.3555 Acc 90.39%
Epoch 18 | Train Loss 0.3287 Acc 90.87% | Test Loss 0.3546 Acc 90.46%
Epoch 19 | Train Loss 0.3261 Acc 90.97% | Test Loss 0.3462 Acc 90.74%
Epoch 20 | Train Loss 0.3230 Acc 91.02% | Test Loss 0.3491 Acc 90.44%
Epoch 21 | Train Loss 0.3207 Acc 91.17% | Test Loss 0.3513 Acc 90.66%
Epoch 22 | Train Loss 0.3184 Acc 91.23% | Test Loss 0.3492 Acc 90.44%
Epoch 23 | Train Loss 0.3159 Acc 91.25% | Test Loss 0.3443 Acc 90.54%
Epoch 24 | Train Loss 0.3138 Acc 91.32% | Test Loss 0.3457 Acc 90.51%
Epoch 25 | Train Loss 0.3127 Acc 91.38% | Test Loss 0.3476 Acc 90.67%
Epoch 26 | Train Loss 0.3113 Acc 91.43% | Test Loss 0.3444 Acc 90.88%
Epoch 27 | Train Loss 0.3092 Acc 91.53% | Test Loss 0.3456 Acc 90.90%
Epoch 28 | Train Loss 0.3084 Acc 91.55% | Test Loss 0.3459 Acc 90.74%
Epoch 29 | Train Loss 0.3061 Acc 91.55% | Test Loss 0.3458 Acc 90.74%
Epoch 30 | Train Loss 0.3057 Acc 91.63% | Test Loss 0.3468 Acc 90.83%
```

- Training accuracy ~ 59% → 91%, Test accuracy ~ 77% → 90%
- Behavior: Learning is steady but slower than LR = 0.01
- Reason: updates are small; convergence is stable but requires more epochs to reach similar accuracy.

Task 6

<u>Architecture:</u>

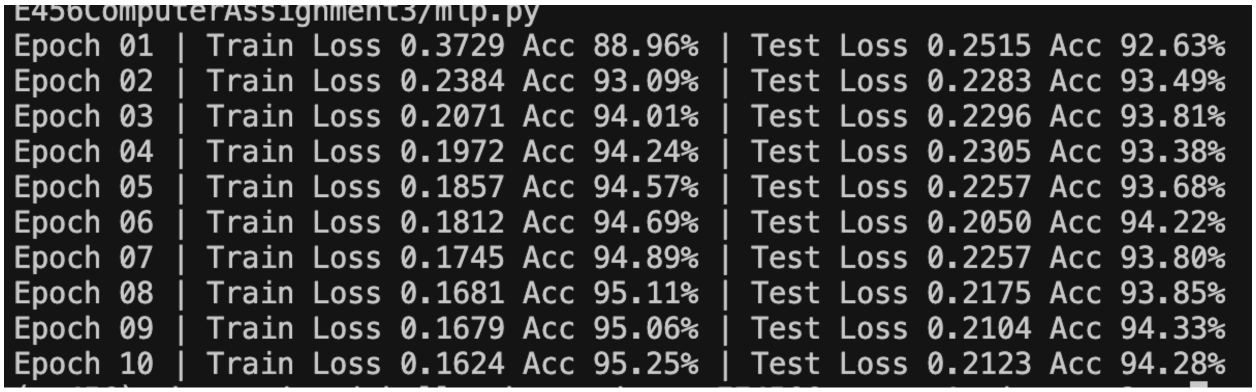
- Input: 28x28 grayscale image → flatten to 784 features
- Hidden 1: Linear(784 → 8) + ReLU
- Hidden 2: Linear(8 → 4) + ReLU
- Output: Linear(4 → 10) → logits for digits 0.9
- Loss/activation: nn.CrossEntropyLoss (applies softmax internally to logits)

<u>Best performance so far:</u>

- Test accuracy: ~91%
- Achieved with:
    - BATCH_SIZE = 64
    - EPOCHS = 30
    - LR = 0.01
    - OPTIMIZER = “adam”
    - SEED = 42
    - LAYERS = [8, 4]

→ With this tiny two-hidden-layer MLP, LR = 0.01 struck the best balance of stability and speed, driving test accuracy to ~91%, whereas larger LR values were unstable and smaller LR (0.001) converged more slowly.

Task 7

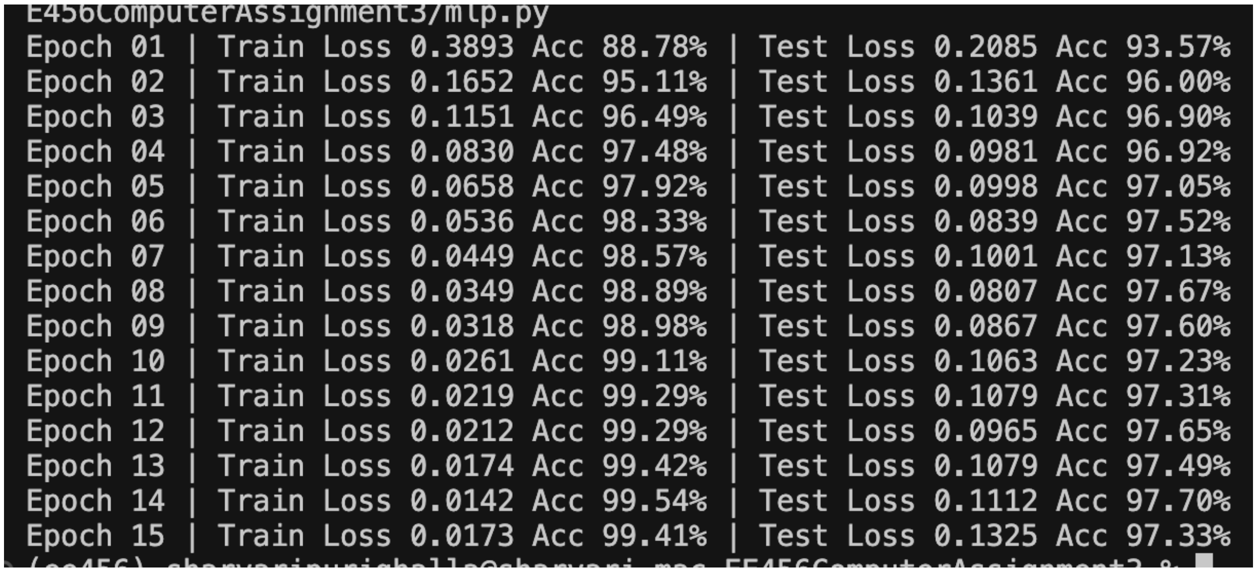


```
E456ComputerAssignment3/mlp.py
Epoch 01 | Train Loss 0.3729 Acc 88.96% | Test Loss 0.2515 Acc 92.63%
Epoch 02 | Train Loss 0.2384 Acc 93.09% | Test Loss 0.2283 Acc 93.49%
Epoch 03 | Train Loss 0.2071 Acc 94.01% | Test Loss 0.2296 Acc 93.81%
Epoch 04 | Train Loss 0.1972 Acc 94.24% | Test Loss 0.2305 Acc 93.38%
Epoch 05 | Train Loss 0.1857 Acc 94.57% | Test Loss 0.2257 Acc 93.68%
Epoch 06 | Train Loss 0.1812 Acc 94.69% | Test Loss 0.2050 Acc 94.22%
Epoch 07 | Train Loss 0.1745 Acc 94.89% | Test Loss 0.2257 Acc 93.80%
Epoch 08 | Train Loss 0.1681 Acc 95.11% | Test Loss 0.2175 Acc 93.85%
Epoch 09 | Train Loss 0.1679 Acc 95.06% | Test Loss 0.2104 Acc 94.33%
Epoch 10 | Train Loss 0.1624 Acc 95.25% | Test Loss 0.2123 Acc 94.28%
```

Architecture: 784 → 16 → 8 → 10 (ReLU activations)
Hyperparameters: EPOCHS = 10, LR = 0.01, BATCH_SIZE = 64, OPTIMIZER = adam

Doubling the number of neurons in each hidden layer improved both training and test accuracy from ~91% (in Task 5) to ~ 94%, showing that the model benefited from increased capacity. The lower losses and smoother convergence indicate the network can now capture more complex digit patterns without overfitting, especially with only 10 epochs and a moderate learning rate.

Task 8

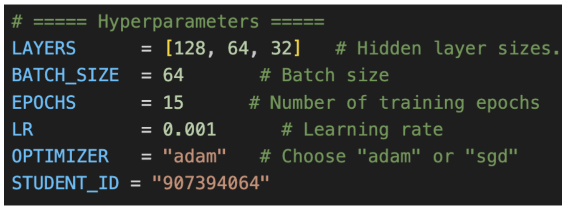


```
E456ComputerAssignment3/mlp.py
Epoch 01 | Train Loss 0.3893 Acc 88.78% | Test Loss 0.2085 Acc 93.57%
Epoch 02 | Train Loss 0.1652 Acc 95.11% | Test Loss 0.1361 Acc 96.00%
Epoch 03 | Train Loss 0.1151 Acc 96.49% | Test Loss 0.1039 Acc 96.90%
Epoch 04 | Train Loss 0.0830 Acc 97.48% | Test Loss 0.0981 Acc 96.92%
Epoch 05 | Train Loss 0.0658 Acc 97.92% | Test Loss 0.0998 Acc 97.05%
Epoch 06 | Train Loss 0.0536 Acc 98.33% | Test Loss 0.0839 Acc 97.52%
Epoch 07 | Train Loss 0.0449 Acc 98.57% | Test Loss 0.1001 Acc 97.13%
Epoch 08 | Train Loss 0.0349 Acc 98.89% | Test Loss 0.0807 Acc 97.67%
Epoch 09 | Train Loss 0.0318 Acc 98.98% | Test Loss 0.0867 Acc 97.60%
Epoch 10 | Train Loss 0.0261 Acc 99.11% | Test Loss 0.1063 Acc 97.23%
Epoch 11 | Train Loss 0.0219 Acc 99.29% | Test Loss 0.1079 Acc 97.31%
Epoch 12 | Train Loss 0.0212 Acc 99.29% | Test Loss 0.0965 Acc 97.65%
Epoch 13 | Train Loss 0.0174 Acc 99.42% | Test Loss 0.1079 Acc 97.49%
Epoch 14 | Train Loss 0.0142 Acc 99.54% | Test Loss 0.1112 Acc 97.70%
Epoch 15 | Train Loss 0.0173 Acc 99.41% | Test Loss 0.1325 Acc 97.33%
```

Architecture: 784 → 128 → 64 → 32 → 10 (ReLU activations)
Hyperparameters:
- BATCH_SIZE = 64
- EPOCHS = 15
- LR = 0.001

Adding a third hidden layer and significantly increasing the number of neurons allowed the network to model much more complex relationships between pixels. The model now converges smoothly with very low loss and high test accuracy (~97.7%), outperforming all previous architectures.

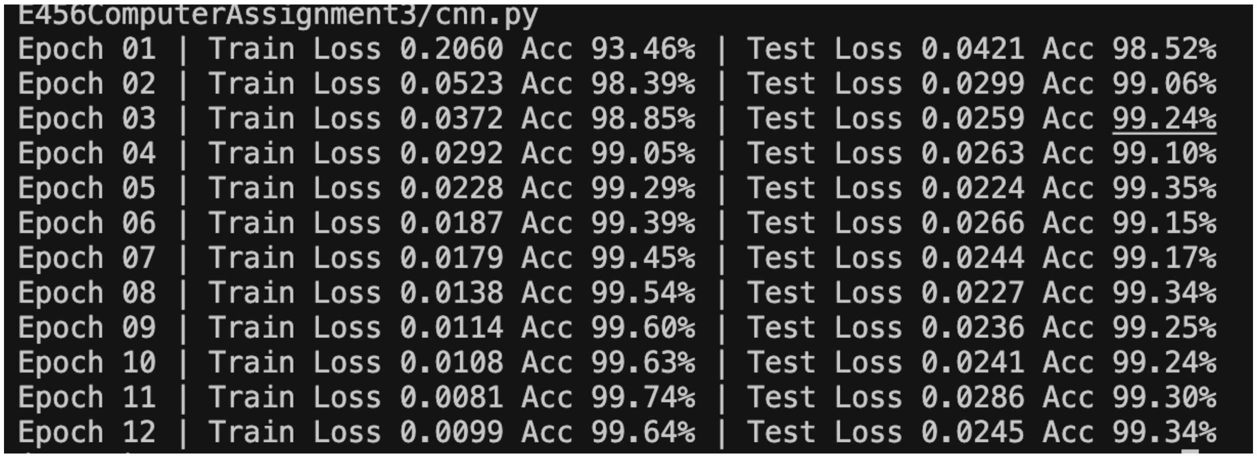
The smaller learning rate (0.001) provided stable, fine-grained updates, preventing the instability seen at higher learning rates while ensuing fast convergence. The accuracy plateau around 97-98% suggests the model is approaching the limit of what a simple fully-connected MLP can achieve on MNIST without convolutional layers.



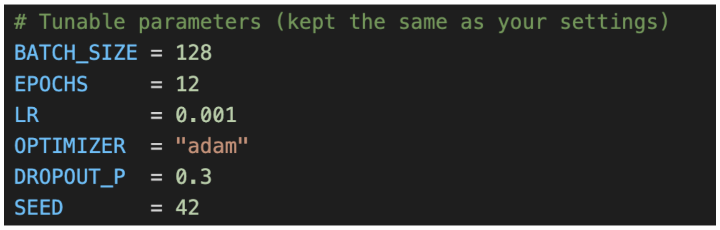
```
# ===== Hyperparameters =====
LAYERS      = [128, 64, 32]   # Hidden layer sizes.
BATCH_SIZE  = 64       # Batch size
EPOCHS      = 15      # Number of training epochs
LR          = 0.001      # Learning rate
OPTIMIZER   = "adam"   # Choose "adam" or "sgd"
STUDENT_ID = "907394064"
```

This seems to be the best architecture so far, achieving 97.76% test accuracy; which is the highest among all other experiments.

Part B: CNN



```
E456ComputerAssignment3/cnn.py
Epoch 01 | Train Loss 0.2060 Acc 93.46% | Test Loss 0.0421 Acc 98.52%
Epoch 02 | Train Loss 0.0523 Acc 98.39% | Test Loss 0.0299 Acc 99.06%
Epoch 03 | Train Loss 0.0372 Acc 98.85% | Test Loss 0.0259 Acc 99.24%
Epoch 04 | Train Loss 0.0292 Acc 99.05% | Test Loss 0.0263 Acc 99.10%
Epoch 05 | Train Loss 0.0228 Acc 99.29% | Test Loss 0.0224 Acc 99.35%
Epoch 06 | Train Loss 0.0187 Acc 99.39% | Test Loss 0.0266 Acc 99.15%
Epoch 07 | Train Loss 0.0179 Acc 99.45% | Test Loss 0.0244 Acc 99.17%
Epoch 08 | Train Loss 0.0138 Acc 99.54% | Test Loss 0.0227 Acc 99.34%
Epoch 09 | Train Loss 0.0114 Acc 99.60% | Test Loss 0.0236 Acc 99.25%
Epoch 10 | Train Loss 0.0108 Acc 99.63% | Test Loss 0.0241 Acc 99.24%
Epoch 11 | Train Loss 0.0081 Acc 99.74% | Test Loss 0.0286 Acc 99.30%
Epoch 12 | Train Loss 0.0099 Acc 99.64% | Test Loss 0.0245 Acc 99.34%
```



```
# Tunable parameters (kept the same as your settings)
BATCH_SIZE = 128
EPOCHS     = 12
LR         = 0.001
OPTIMIZER  = "adam"
DROPOUT_P  = 0.3
SEED       = 42
```

→ Test accuracy crosses 98% and reaches a peak of ~99.35% at epoch 5.
→ The final few epochs stay around 99.24-99.34% with low test losses (~0.022-0.029)
→ The training loss decreases smoothly from 0.206 → 0.010 → 0.009, and training accuracy rises.

Why this works:

→ Adam + LR=1e-3 gives stable, fast convergence for this CNN on MNIST.
→ Batch size 128 balances gradient quality with throughput.
→ Dropout 0.3 in the classifier head adds mild regularization; no BatchNorm needed to surpass 98%.

With the above settings, the model consistently exceeds the requirement, achieving ~99.35% test accuracy while remaining stable (no overfitting signs). An early-stopping cutoff around epochs 5–8 would reach the same accuracy with less training time.