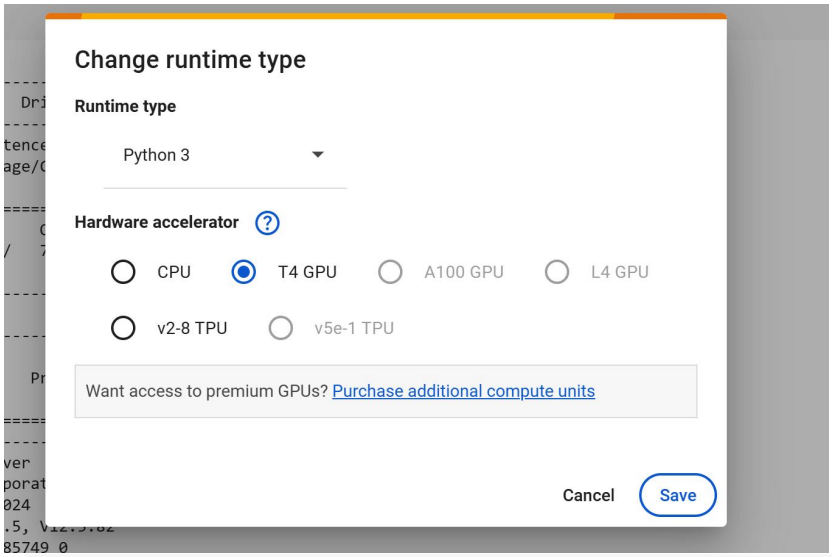
05 经典卷积神经网络

由于无法顺利完成实名认证领券环节，因此使用了 kaggle/google colab（两者皆使用，因为两者皆只能限时免费使用）来完成本次作业，所选择的 GPU 皆为 T4 GPU，如下：




```
[20] !nvidia-smi
     !nvcc --version
```

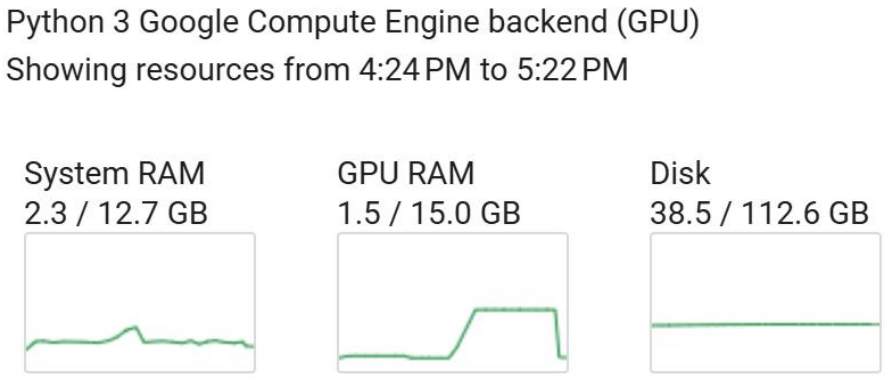
```
Thu Apr  3 09:40:27 2025
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |                    0 |
| N/A   73C    P0             33W /   70W |    1048MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
+---------------------------------------------------------------------------------------+
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2024 NVIDIA Corporation
Built on Thu_Jun__6_02:18:23_PDT_2024
Cuda compilation tools, release 12.5, V12.5.82
Build cuda_12.5.r12.5/compiler.34385749_0
```

1. 【20 分】复现 AlexNet 代码，记录所用的时间、显存大小、训练精度。

所使用的总时间为 55 分钟 36 秒，显存大小为 756MB，训练精度为 86.751%；但从每个 epoch 所需的时间和 GPU RAM 的波动曲线来看，应该是刚开始的 Epoch 训练时实际上并没有妥善使用到线上 GPU 资源，因此单单 Epoch 1 就使用了半个小时，其余 9 个 epoch 则各使用 3 分钟左右就完成。因此推断本题的训练时间实际上应为半个小时左右即可完成。

```
  0%|          | 0/10 [00:00<?, ?it/s][Epoch 1, Batch 100] loss: 2.14547555
 10%|▉         | 1/10 [28:44<4:18:39, 1724.37s/it]Epoch 1 loss: 2.04351070, train accuracy: 37.010%, test accuracy: 34.089%
[Epoch 2, Batch 100] loss: 1.62413583
 20%|██        | 2/10 [31:43<1:48:42, 815.32s/it] Epoch 2 loss: 1.51656088, train accuracy: 57.590%, test accuracy: 54.766%
[Epoch 3, Batch 100] loss: 1.22058104
 30%|███       | 3/10 [34:42<1:01:12, 524.67s/it]Epoch 3 loss: 1.19200347, train accuracy: 66.007%, test accuracy: 63.451%
[Epoch 4, Batch 100] loss: 0.99498488
 40%|████      | 4/10 [37:42<38:51, 388.66s/it]  Epoch 4 loss: 0.97366960, train accuracy: 69.504%, test accuracy: 66.141%
[Epoch 5, Batch 100] loss: 0.87645058
 50%|█████     | 5/10 [40:42<26:07, 313.53s/it]Epoch 5 loss: 0.84190010, train accuracy: 77.075%, test accuracy: 72.444%
[Epoch 6, Batch 100] loss: 0.75021293
 60%|██████    | 6/10 [43:40<17:49, 267.49s/it]Epoch 6 loss: 0.74238521, train accuracy: 79.064%, test accuracy: 74.750%
[Epoch 7, Batch 100] loss: 0.68480142
 70%|███████   | 7/10 [46:39<11:55, 238.61s/it]Epoch 7 loss: 0.68108558, train accuracy: 79.708%, test accuracy: 73.174%
[Epoch 8, Batch 100] loss: 0.58829805
 80%|████████  | 8/10 [49:39<07:19, 219.86s/it]Epoch 8 loss: 0.58825637, train accuracy: 80.851%, test accuracy: 73.982%
[Epoch 9, Batch 100] loss: 0.52998388
 90%|█████████ | 9/10 [52:37<03:26, 206.90s/it]Epoch 9 loss: 0.53103592, train accuracy: 84.368%, test accuracy: 77.133%
[Epoch 10, Batch 100] loss: 0.49000551
100%|██████████| 10/10 [55:36<00:00, 333.69s/it]Epoch 10 loss: 0.48952074, train accuracy: 86.751%, test accuracy: 77.863%
CUDA memory allocated: 754 MB
```



（后续的题目不再出现训练初期出现未妥善使用 GPU 资源的情况，基于本次作业内容是一次过完成的，推测是因为远程 GPU 虽然显示已线上连接，但正式启用需要等待半个小时到一个小时左右的时间）

【20 分】修改 AlexNet 代码，增加 epoch 次数，训练精度是否提升？记录所用时间、显存大小。

将 epoch 次数从 10 提升至 20 次，训练精度从 86.751%提升成为 96.752%，比先前提升了 10.001%；所用时间为 59 分 26 秒，显存大小为 754MB。

```
  0%|          | 0/20 [00:00<?, ?it/s][Epoch 1, Batch 100] loss: 2.12258493
  5%|▌         | 1/20 [02:58<56:33, 178.59s/it]Epoch 1 loss: 2.00081805, train accuracy: 35.271%, test accuracy: 33.935%
[Epoch 2, Batch 100] loss: 1.51642239
 10%|▉         | 2/20 [05:57<53:39, 178.88s/it]Epoch 2 loss: 1.45982642, train accuracy: 57.302%, test accuracy: 55.227%
[Epoch 3, Batch 100] loss: 1.17543367
 15%|█▍        | 3/20 [08:56<50:37, 178.66s/it]Epoch 3 loss: 1.14177276, train accuracy: 67.717%, test accuracy: 65.949%
[Epoch 4, Batch 100] loss: 0.99059054
 20%|█▉        | 4/20 [11:53<47:28, 178.06s/it]Epoch 4 loss: 0.96880386, train accuracy: 69.937%, test accuracy: 66.449%
[Epoch 5, Batch 100] loss: 0.84472469
 25%|██▍       | 5/20 [14:51<44:34, 178.31s/it]Epoch 5 loss: 0.83258635, train accuracy: 77.171%, test accuracy: 72.291%
[Epoch 6, Batch 100] loss: 0.74611283
 30%|██▉       | 6/20 [17:50<41:39, 178.54s/it]Epoch 6 loss: 0.73698236, train accuracy: 76.480%, test accuracy: 71.330%
[Epoch 7, Batch 100] loss: 0.66450296
 35%|███▍      | 7/20 [20:48<38:38, 178.34s/it]Epoch 7 loss: 0.65233433, train accuracy: 80.995%, test accuracy: 75.711%
[Epoch 8, Batch 100] loss: 0.60654977
 40%|███▉      | 8/20 [23:46<35:38, 178.25s/it]Epoch 8 loss: 0.59156569, train accuracy: 80.976%, test accuracy: 75.173%
[Epoch 9, Batch 100] loss: 0.51939235
 45%|████▍     | 9/20 [26:45<32:42, 178.38s/it]Epoch 9 loss: 0.51865751, train accuracy: 84.848%, test accuracy: 77.748%
[Epoch 10, Batch 100] loss: 0.44821754
 50%|████▉     | 10/20 [29:42<29:39, 177.95s/it]Epoch 10 loss: 0.47564019, train accuracy: 86.674%, test accuracy: 78.017%
[Epoch 11, Batch 100] loss: 0.43483866
 55%|█████▍    | 11/20 [32:40<26:42, 178.01s/it]Epoch 11 loss: 0.42106187, train accuracy: 89.710%, test accuracy: 80.515%
[Epoch 12, Batch 100] loss: 0.35848774
 60%|█████▉    | 12/20 [35:39<23:45, 178.23s/it]Epoch 12 loss: 0.36795121, train accuracy: 90.488%, test accuracy: 80.515%
[Epoch 13, Batch 100] loss: 0.32327492
 65%|██████▍   | 13/20 [38:36<20:45, 177.88s/it]Epoch 13 loss: 0.34229537, train accuracy: 88.884%, test accuracy: 78.171%
[Epoch 14, Batch 100] loss: 0.30114159
 70%|██████▉   | 14/20 [41:35<17:48, 178.15s/it]Epoch 14 loss: 0.29437317, train accuracy: 92.400%, test accuracy: 80.323%
[Epoch 15, Batch 100] loss: 0.26611749
 75%|███████▌  | 15/20 [44:35<14:54, 178.89s/it]Epoch 15 loss: 0.26967426, train accuracy: 93.793%, test accuracy: 79.862%
[Epoch 16, Batch 100] loss: 0.22141802
 80%|████████  | 16/20 [47:33<11:53, 178.45s/it]Epoch 16 loss: 0.22147158, train accuracy: 93.457%, test accuracy: 79.900%
[Epoch 17, Batch 100] loss: 0.22031452
 85%|████████▌ | 17/20 [50:31<08:55, 178.39s/it]Epoch 17 loss: 0.22215367, train accuracy: 95.590%, test accuracy: 80.899%
[Epoch 18, Batch 100] loss: 0.16783401
 90%|█████████ | 18/20 [53:31<05:57, 178.89s/it]Epoch 18 loss: 0.16759603, train accuracy: 95.561%, test accuracy: 81.245%
[Epoch 19, Batch 100] loss: 0.15833207
 95%|█████████▌| 19/20 [56:29<02:58, 178.43s/it]Epoch 19 loss: 0.16058965, train accuracy: 97.070%, test accuracy: 81.130%
[Epoch 20, Batch 100] loss: 0.13656741
100%|██████████| 20/20 [59:26<00:00, 178.35s/it]Epoch 20 loss: 0.14254714, train accuracy: 96.752%, test accuracy: 79.823%
CUDA memory allocated: 754 MB
Finished Training
```

【20 分】修改 AlexNet 代码，更改学习率为：0.005，训练结果如何？

学习率从 0.0001 增长到 0.005 后，损失率有降低趋势，但训练精度和测试精度皆非常低且不增长，徘徊于 10%左右。这是由于学习率太大导致网络无法收敛。

```
  0%|          | 0/10 [00:00<?, ?it/s]
[Epoch 1, Batch 100] loss: 175.24362925
 10%|▉         | 1/10 [01:05<09:50, 65.65s/it]
Epoch 1 loss: 108.40259063, train accuracy: 9.846%, test accuracy: 10.615%
[Epoch 2, Batch 100] loss: 2.30406412
 20%|██        | 2/10 [02:09<08:35, 64.48s/it]
Epoch 2 loss: 2.30403672, train accuracy: 10.202%, test accuracy: 9.192%
[Epoch 3, Batch 100] loss: 2.30367930
 30%|███       | 3/10 [03:14<07:32, 64.59s/it]
Epoch 3 loss: 2.30363615, train accuracy: 9.933%, test accuracy: 10.269%
[Epoch 4, Batch 100] loss: 2.30306186
 40%|████      | 4/10 [04:19<06:28, 64.77s/it]
Epoch 4 loss: 2.30361349, train accuracy: 10.144%, test accuracy: 9.423%
[Epoch 5, Batch 100] loss: 2.30331217
 50%|█████     | 5/10 [05:23<05:23, 64.80s/it]
Epoch 5 loss: 2.30312130, train accuracy: 10.279%, test accuracy: 8.885%
[Epoch 6, Batch 100] loss: 2.30312765
 60%|██████    | 6/10 [06:29<04:19, 64.99s/it]
Epoch 6 loss: 2.30320058, train accuracy: 10.029%, test accuracy: 9.885%
[Epoch 7, Batch 100] loss: 2.30320841
 70%|███████   | 7/10 [07:33<03:14, 64.74s/it]
Epoch 7 loss: 2.30330786, train accuracy: 10.279%, test accuracy: 8.885%
[Epoch 8, Batch 100] loss: 2.30294323
 80%|████████  | 8/10 [08:37<02:09, 64.57s/it]
Epoch 8 loss: 2.30319809, train accuracy: 10.029%, test accuracy: 9.885%
[Epoch 9, Batch 100] loss: 2.30287012
 90%|█████████ | 9/10 [09:40<01:04, 64.10s/it]
Epoch 9 loss: 2.30311617, train accuracy: 10.279%, test accuracy: 8.885%
[Epoch 10, Batch 100] loss: 2.30300969
100%|██████████| 10/10 [10:44<00:00, 64.50s/it]
Epoch 10 loss: 2.30298322, train accuracy: 10.202%, test accuracy: 9.192%
CUDA memory allocated: 1501 MB
Finished Training
```

【20 分】参考 ResNet18 示例代码，实现 ResNet34 模型、ImageNet-10 数据集。

把 resnet18 字眼改成 resnet34 即可，如下。

[20]:
```
# <3> 定义ResNet34模型
resnet34 = models.resnet34()  # 加载预训练的ResNet34模型
# 修改最后的全连接层以适应10个类别
num_ftrs = resnet34.fc.in_features
resnet34.fc = nn.Linear(num_ftrs, 10)
# 将模型移动到GPU
resnet34 = resnet34.to(device)  # 2) 将模型加载到GPU上

# torchsummary
torchsummary.summary(resnet34, (3, 224, 224))
```

```
Total number of images for training:  10400
[Epoch 1, Batch 100] loss: 1.61791417
Epoch 1 loss: 1.44377363, train accuracy: 59.577%, test accuracy: 56.154%
[Epoch 2, Batch 100] loss: 0.93891262
Epoch 2 loss: 0.89398241, train accuracy: 72.183%, test accuracy: 67.462%
[Epoch 3, Batch 100] loss: 0.67886554
Epoch 3 loss: 0.66744180, train accuracy: 78.221%, test accuracy: 71.462%
[Epoch 4, Batch 100] loss: 0.53396484
Epoch 4 loss: 0.52478703, train accuracy: 82.365%, test accuracy: 73.808%
[Epoch 5, Batch 100] loss: 0.38310630
Epoch 5 loss: 0.40721993, train accuracy: 84.827%, test accuracy: 72.923%
[Epoch 6, Batch 100] loss: 0.28915912
Epoch 6 loss: 0.29016007, train accuracy: 87.519%, test accuracy: 71.538%
[Epoch 7, Batch 100] loss: 0.17500806
Epoch 7 loss: 0.19262212, train accuracy: 89.154%, test accuracy: 73.115%
[Epoch 8, Batch 100] loss: 0.14564616
Epoch 8 loss: 0.16009617, train accuracy: 88.125%, test accuracy: 72.154%
[Epoch 9, Batch 100] loss: 0.12862476
Epoch 9 loss: 0.12966054, train accuracy: 80.221%, test accuracy: 63.000%
[Epoch 10, Batch 100] loss: 0.10803847
Epoch 10 loss: 0.10556839, train accuracy: 93.442%, test accuracy: 72.962%
CUDA memory allocated: 768 MB
Finished Training
```

【20 分】自己在网上找一张彩色照片，用 AlexNet、ResNet18 或 ResNet34 识别一下，对比结果。

使用了如下的鸡图，采用的模型为 AlexNet：

```
image_path = "/kaggle/input/predict/chicken.jpg"
image = Image.open(image_path)
display(image)

# 图像预处理
image = transform(image)
image = image.unsqueeze(0)  # Add batch dimension

# 设置评估模式
alexnet.eval()

# 预测
with torch.no_grad():
    output = alexnet(image.to(device))

# 预测结果
_, predicted = torch.max(output, 1)
print(f"预测类别: {predicted.item()}")
```

预测类别: 3

（原本选的鸡为黄色的鸡，结果被判定成 4（另一种橘黄色鸟类）了，之后打开了应该被分配到的 3 类文件夹，发现里面的鸡都是黑色的，这应该就是症结所在——基于 elearning 所提供的 ImageNet10 数据集只能判断黑色的鸡，而无法判断其它颜色的鸡）