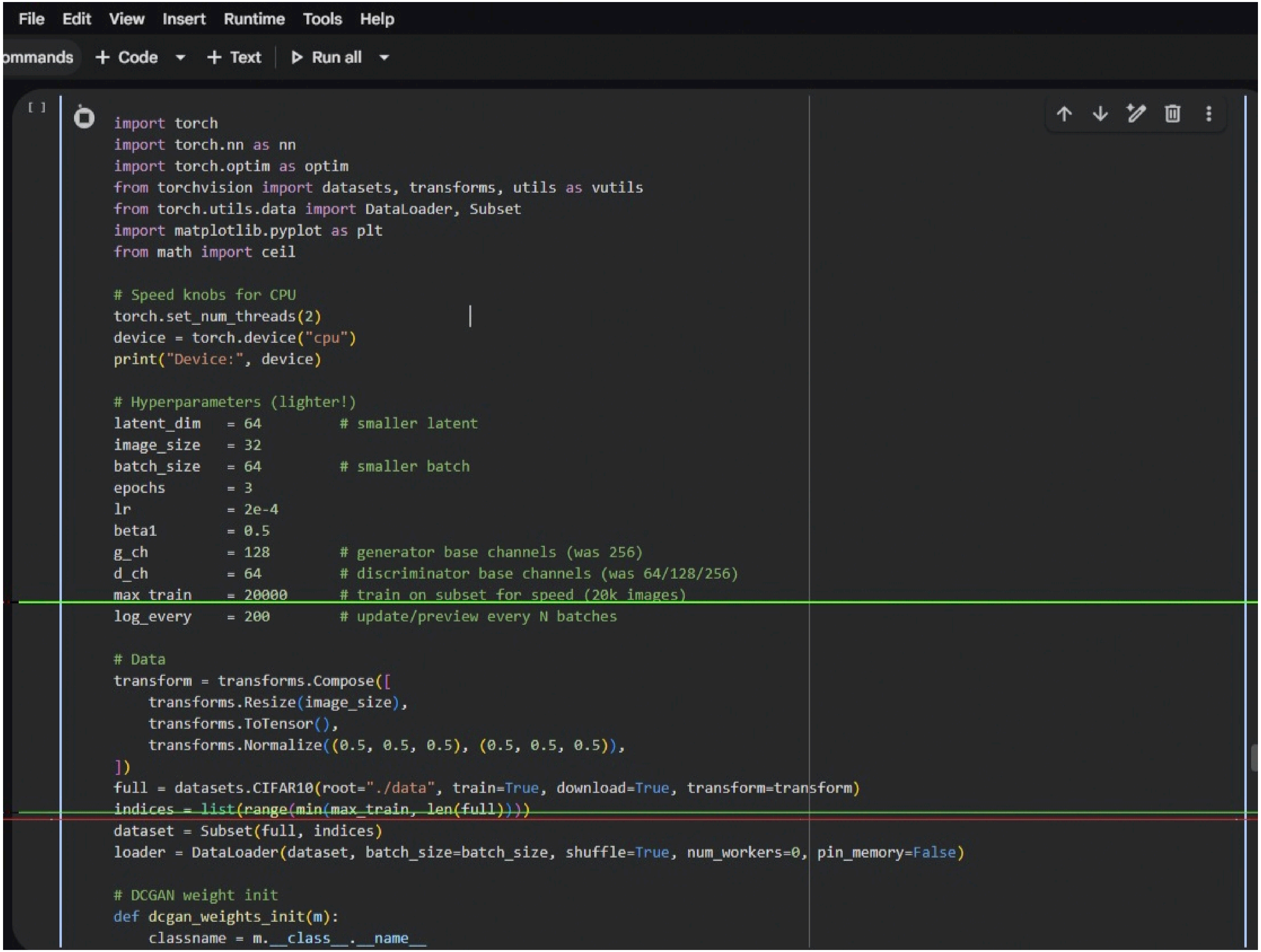
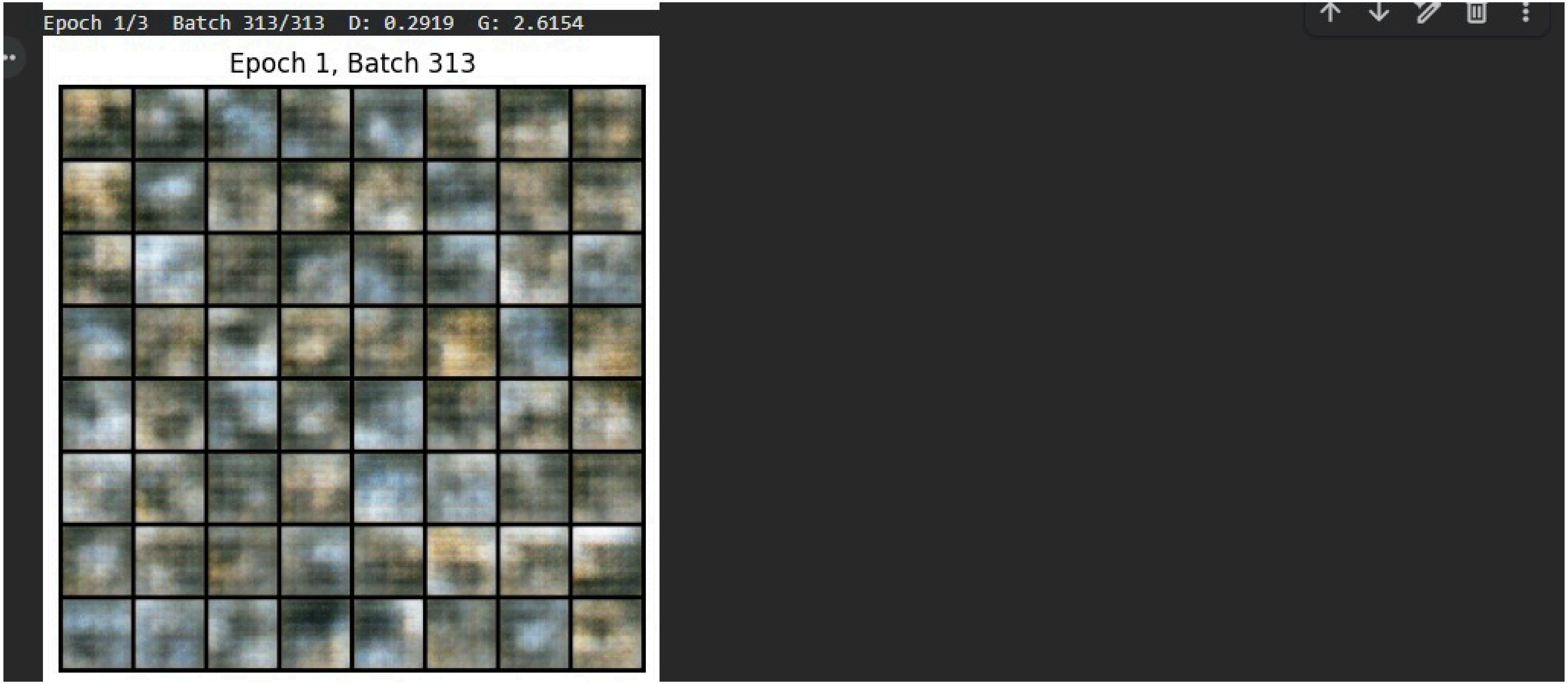
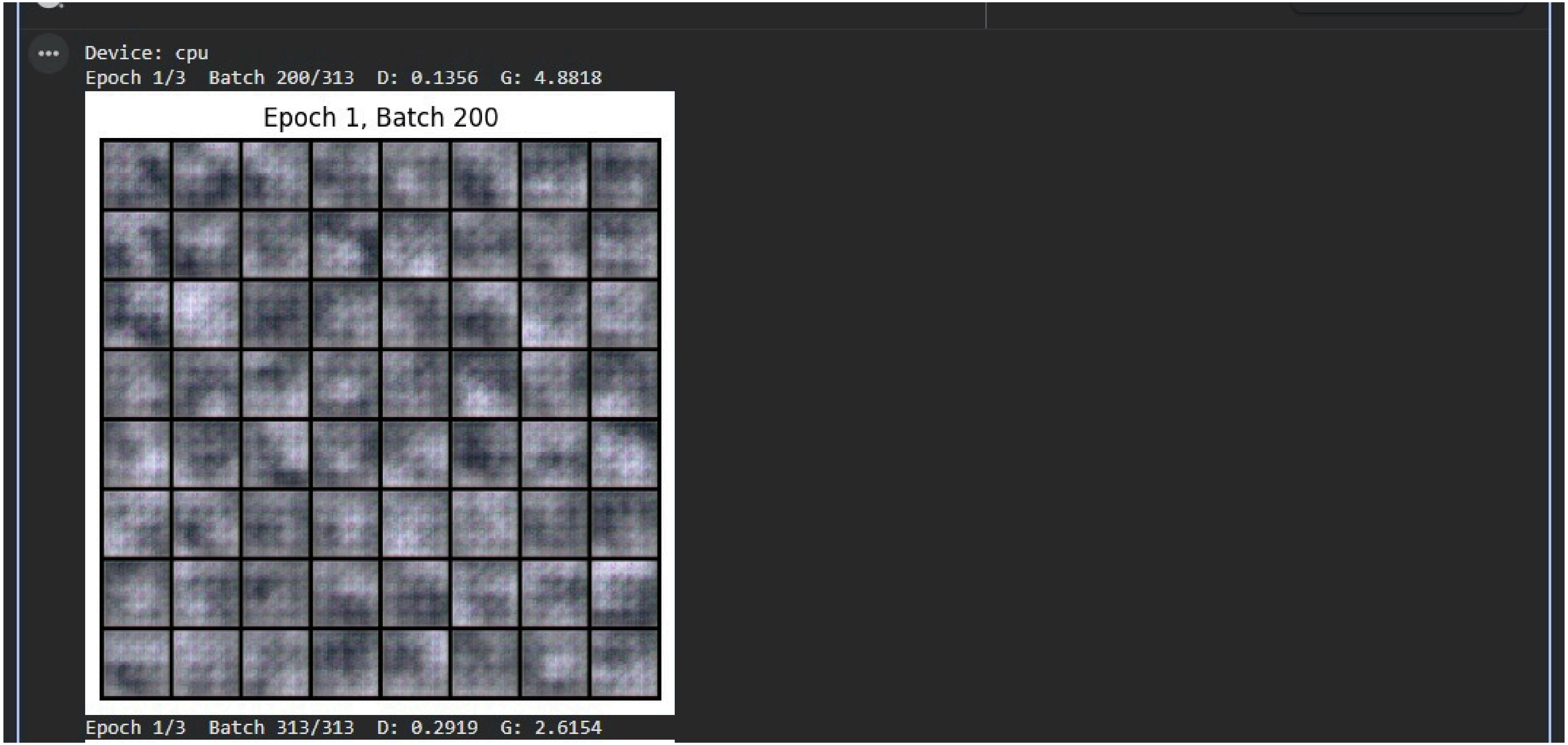


```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms, utils as vutils
from torch.utils.data import DataLoader, Subset
import matplotlib.pyplot as plt
from math import ceil

# Speed knobs for CPU
torch.set_num_threads(2)
device = torch.device("cpu")
print("Device:", device)

# Hyperparameters (lighter!)
latent_dim   = 64         # smaller latent
image_size   = 32
batch_size   = 64         # smaller batch
epochs       = 3
lr           = 2e-4
beta1        = 0.5
g_ch         = 128        # generator base channels (was 256)
d_ch         = 64         # discriminator base channels (was 64/128/256)
max_train    = 20000      # train on subset for speed (20k images)
log_every    = 200        # update/preview every N batches

# Data
transform = transforms.Compose([
    transforms.Resize(image_size),
    transforms.ToTensor(),
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
])
full = datasets.CIFAR10(root="./data", train=True, download=True, transform=transform)
indices = list(range(min(max_train, len(full))))
dataset = Subset(full, indices)
loader = DataLoader(dataset, batch_size=batch_size, shuffle=True, num_workers=0, pin_memory=False)

# DCGAN weight init
def dcgan_weights_init(m):
    classname = m.__class__.__name__
```



```
Epoch 1/3  Batch 313/313  D: 0.2919  G: 2.6154
```



```
Device: cpu
Epoch 1/3  Batch 200/313  D: 0.1356  G: 4.8818
```

```
Epoch 1/3  Batch 313/313  D: 0.2919  G: 2.6154
```