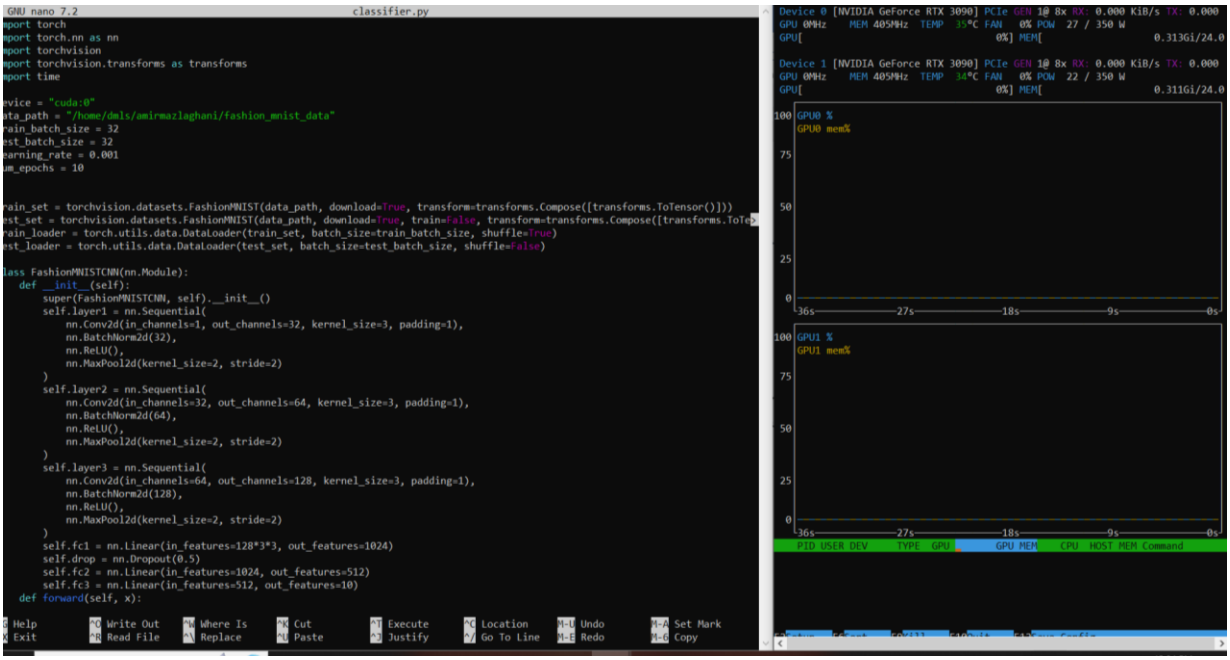
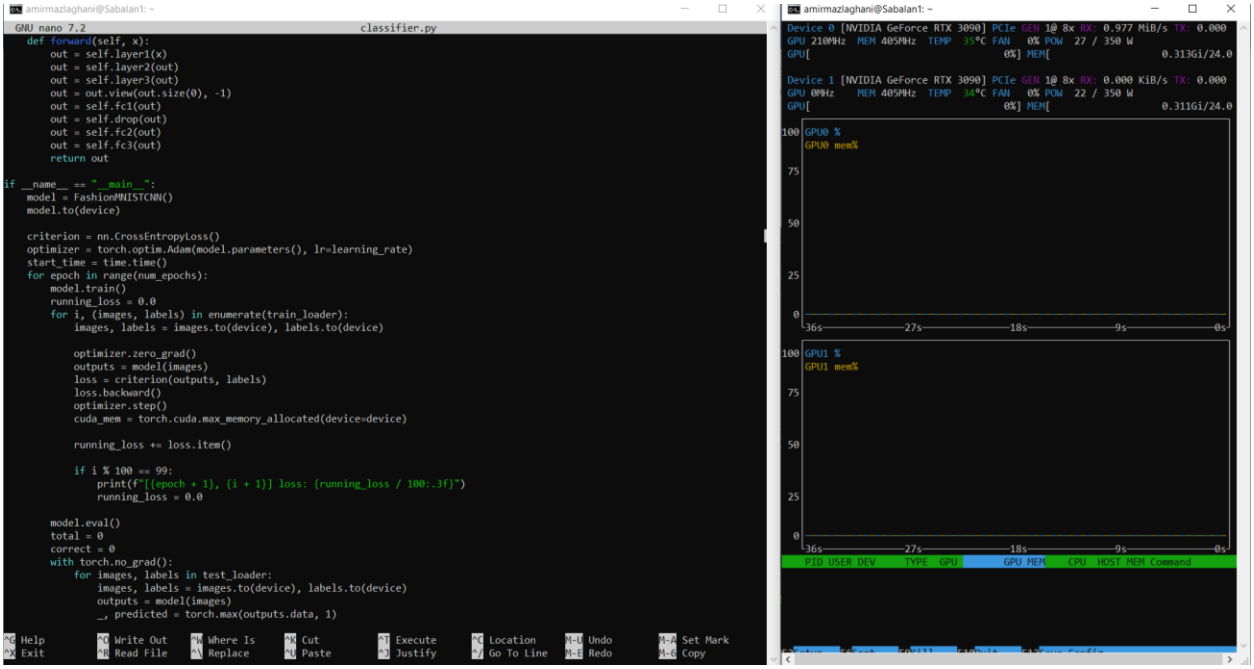
تمرین دو (محمدباربد امیرمزلقانی) – ۸۱۰۱۰۲۳۴۸

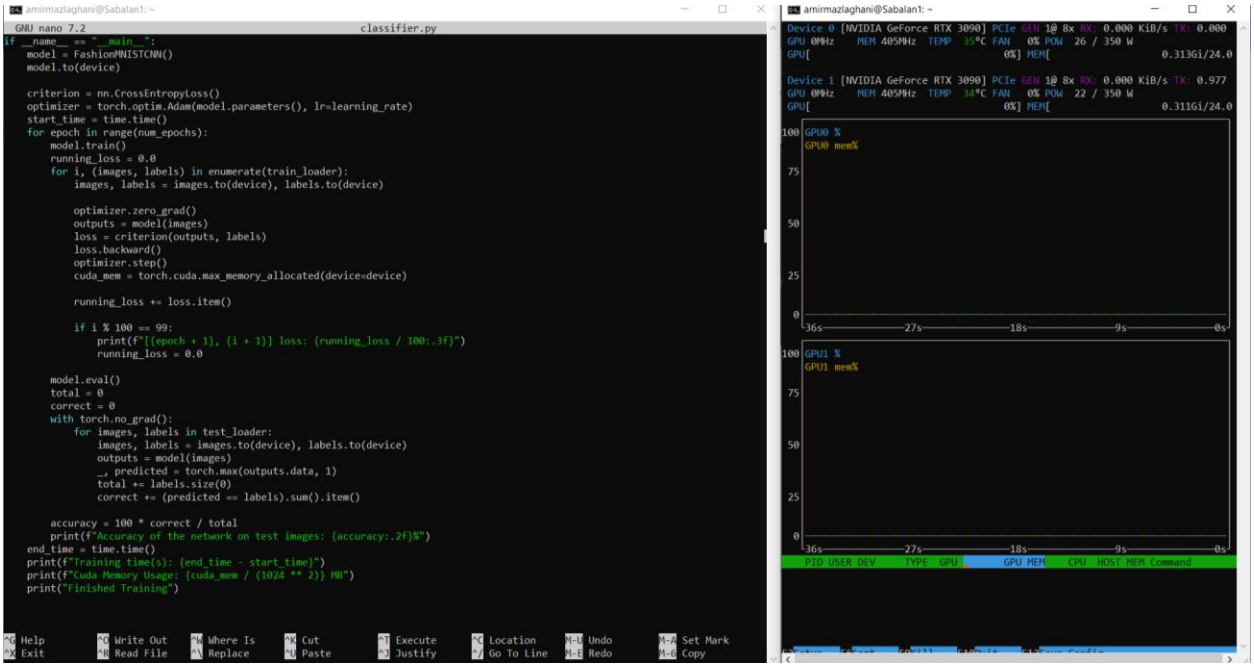
سوال یک:

الف)

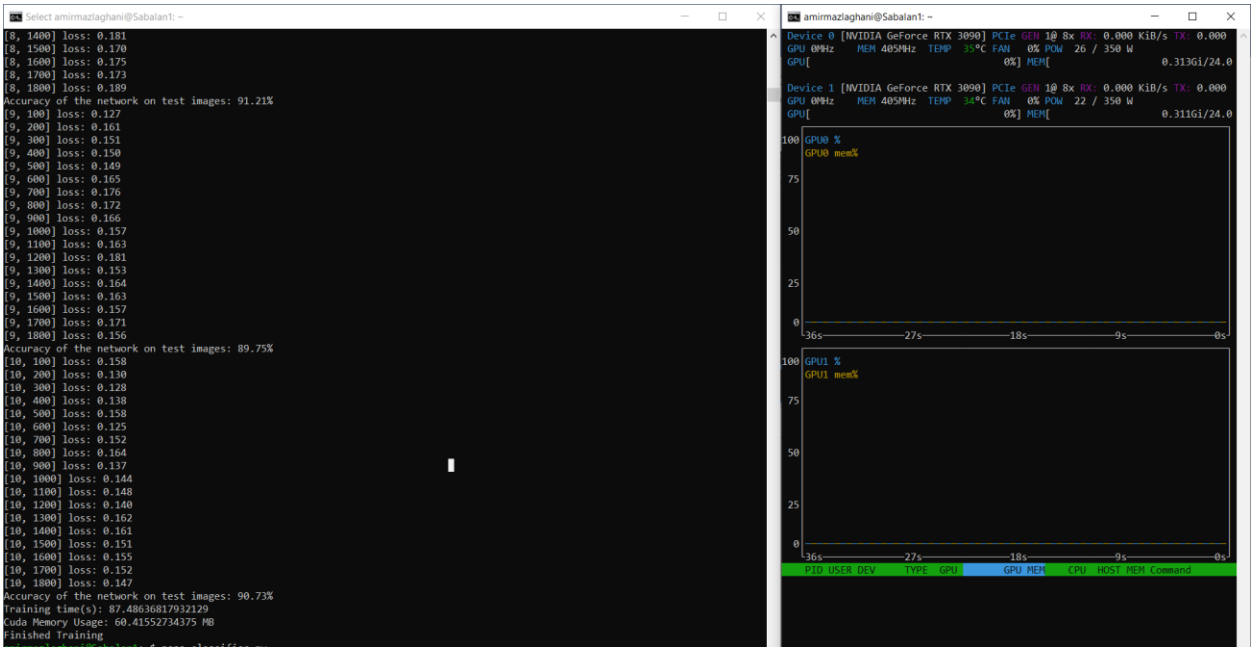
در این سوال بر مبنای کد پیاده سازی شده توسط TA یک سری تغییرات لازم را اعمال میکنیم .(یک لایه کانولوشن اضافه میکنیم، تعداد epoch را به ۱۰ افزایش میدهیم و جهت جلوگیری از بیش‌برازش dropout rate را به ۰.۵ افزایش دادیم)





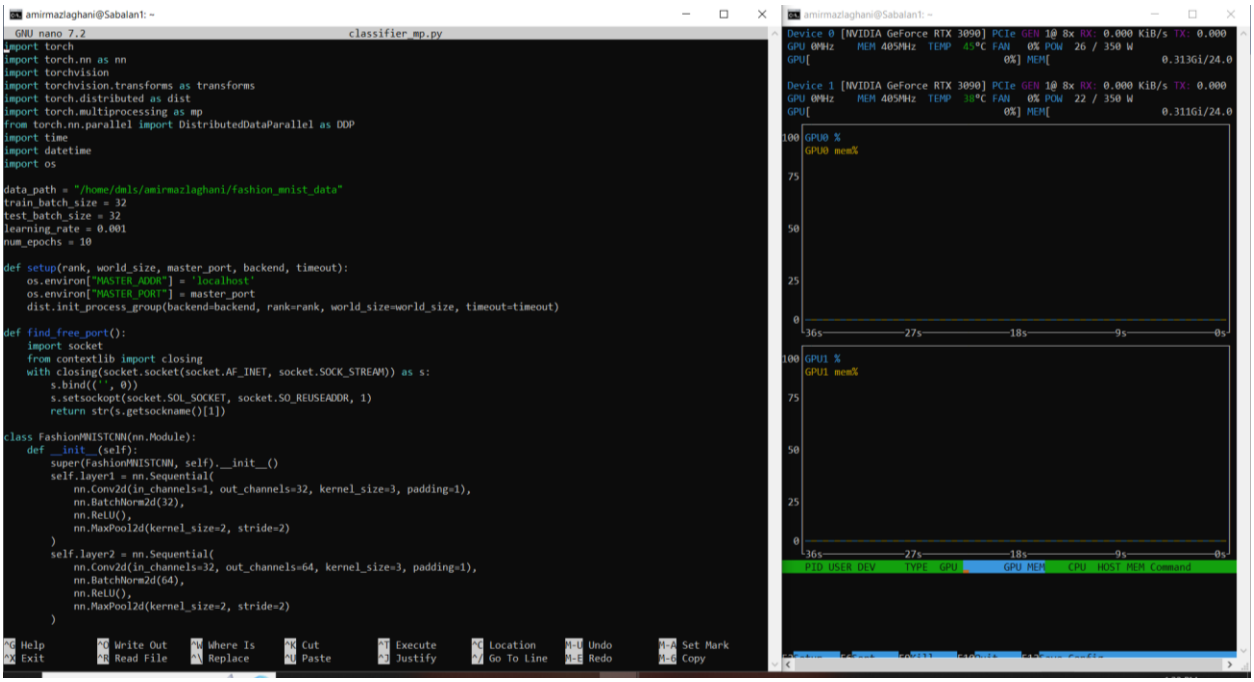


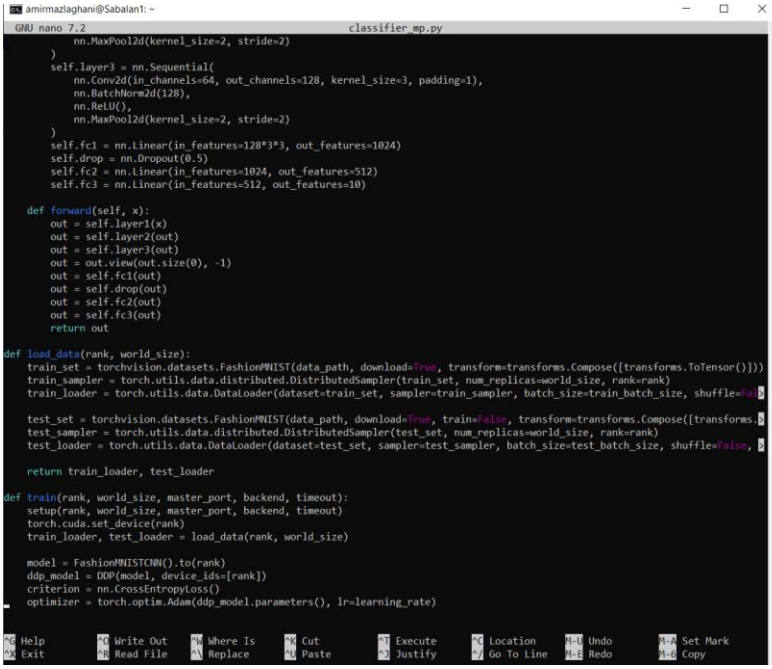
در ادامه دقت و زمان و حافظه را مشاهده میکنیم :

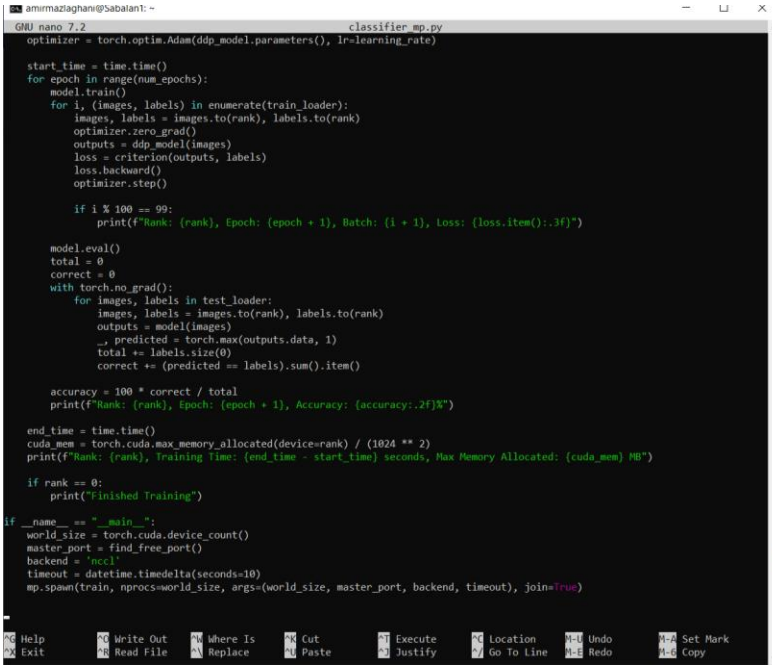


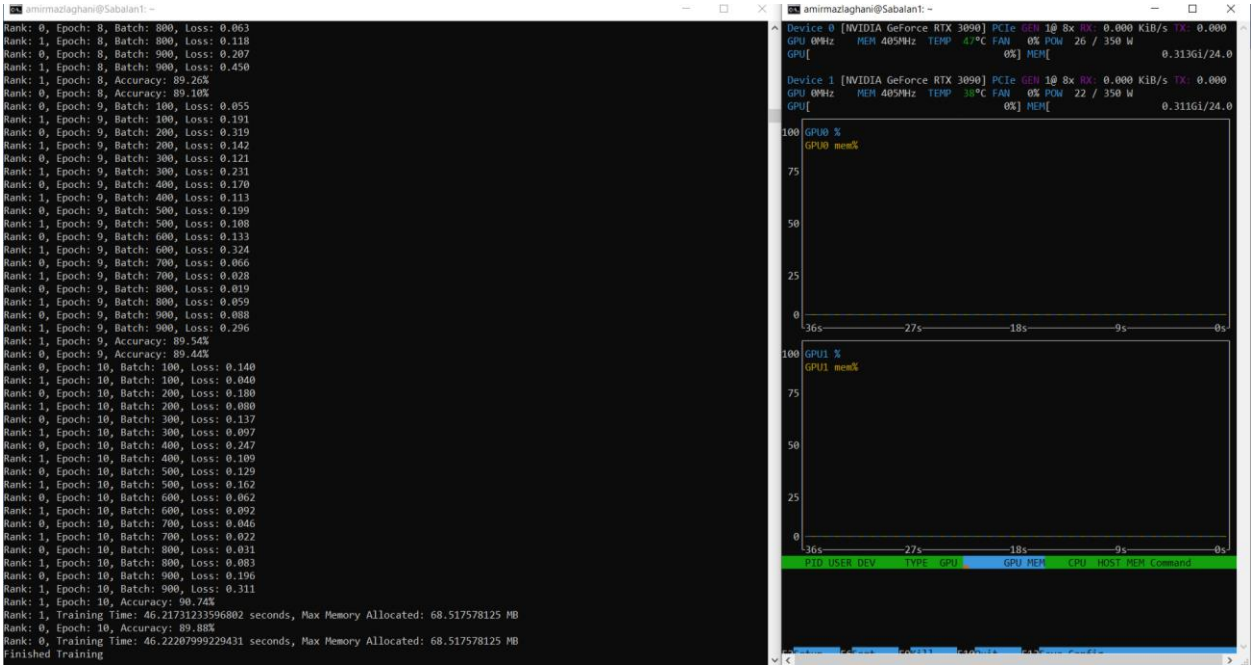
ب)

در قسمت دوم باز مبنا را کد TA قرار میدهیم مدل را هم ، همان مدل مرحله الف میگذاریم در ادامه ابتدا کد این بخش و سپس خروجی آن که نشان دهنده کاهش محسوس زمان است را مشاهده میکنیم.



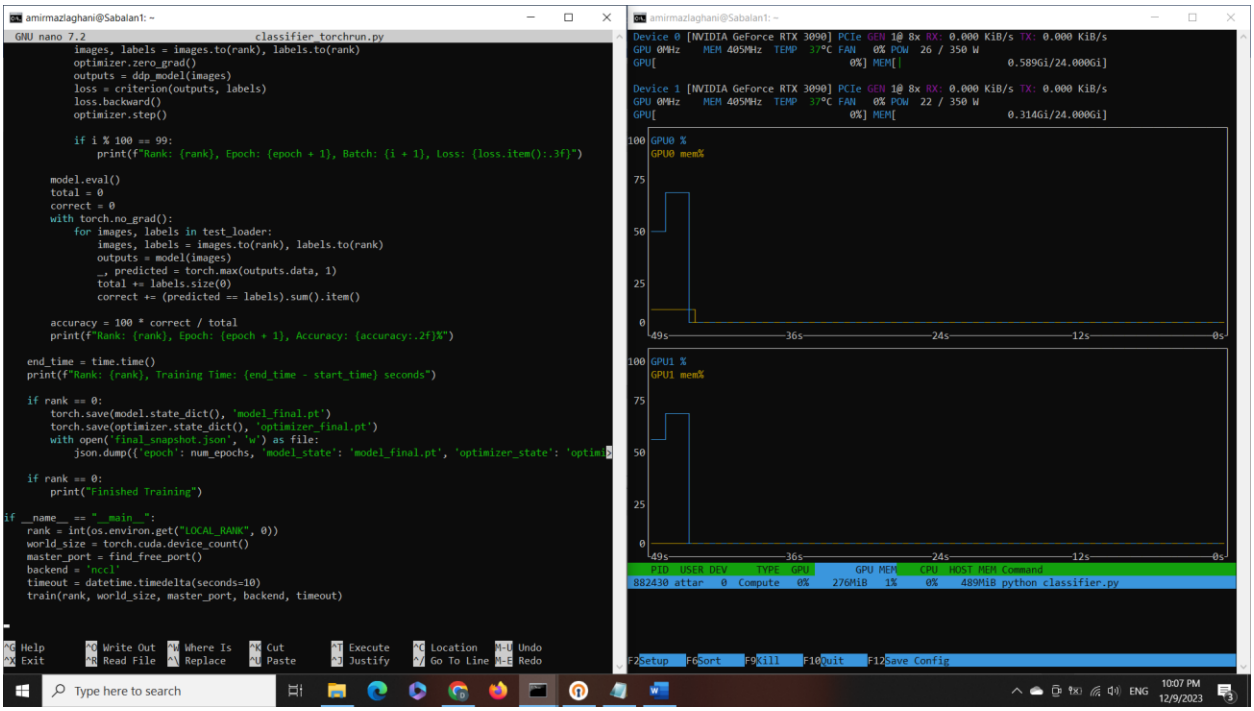

amirmazlaghani@Sabalan1: ~
GNU nano 7.2 classifier_mp.py
nn.MaxPool2d(kernel_size=2, stride=2)
)
self.layer3 = nn.Sequential(
nn.Conv2d(in_channels=64, out_channels=128, kernel_size=3, padding=1),
nn.BatchNorm2d(128),
nn.ReLU(),
nn.MaxPool2d(kernel_size=2, stride=2)
)
self.fc1 = nn.Linear(in_features=128*3*3, out_features=1024)
self.drop = nn.Dropout(0.5)
self.fc2 = nn.Linear(in_features=1024, out_features=512)
self.fc3 = nn.Linear(in_features=512, out_features=10)
def forward(self, x):
out = self.layer1(x)
out = self.layer2(out)
out = self.layer3(out)
out = out.view(out.size(0), -1)
out = self.fc1(out)
out = self.drop(out)
out = self.fc2(out)
out = self.fc3(out)
return out
def load_data(rank, world_size):
train_set = torchvision.datasets.FashionMNIST(data_path, download=True, transform=transforms.Compose([transforms.ToTensor()]))
train_sampler = torch.utils.data.distributed.DistributedSampler(train_set, num_replicas=world_size, rank=rank)
train_loader = torch.utils.data.DataLoader(dataset=train_set, sampler=train_sampler, batch_size=train_batch_size, shuffle=Fal
test_set = torchvision.datasets.FashionMNIST(data_path, download=True, train=False, transform=transforms.Compose([transforms.
test_sampler = torch.utils.data.distributed.DistributedSampler(test_set, num_replicas=world_size, rank=rank)
test_loader = torch.utils.data.DataLoader(dataset=test_set, sampler=test_sampler, batch_size=test_batch_size, shuffle=False,
return train_loader, test_loader
def train(rank, world_size, master_port, backend, timeout):
setup(rank, world_size, master_port, backend, timeout)
torch.cuda.set_device(rank)
train_loader, test_loader = load_data(rank, world_size)
model = FashionMNISTCNN().to(rank)
ddp_model = DDP(model, device_ids=[rank])
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(ddp_model.parameters(), lr=learning_rate)
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy
amirmazlaghani@Sabalan1: ~
Device 0 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000
GPU 0MHz MEM 405MHz TEMP 45°C FAN 0% POW 26 / 350 W
GPU[0%] MEM[0.313Gi/24.0
Device 1 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000
GPU 0MHz MEM 405MHz TEMP 38°C FAN 0% POW 22 / 350 W
GPU[0%] MEM[0.311Gi/24.0
GPU0 %
GPU0 mem%
GPU1 %
GPU1 mem%
36s 27s 18s 9s 0s
PID USER DEV TYPE GPU GPU_MEM CPU HOST_MEM Command

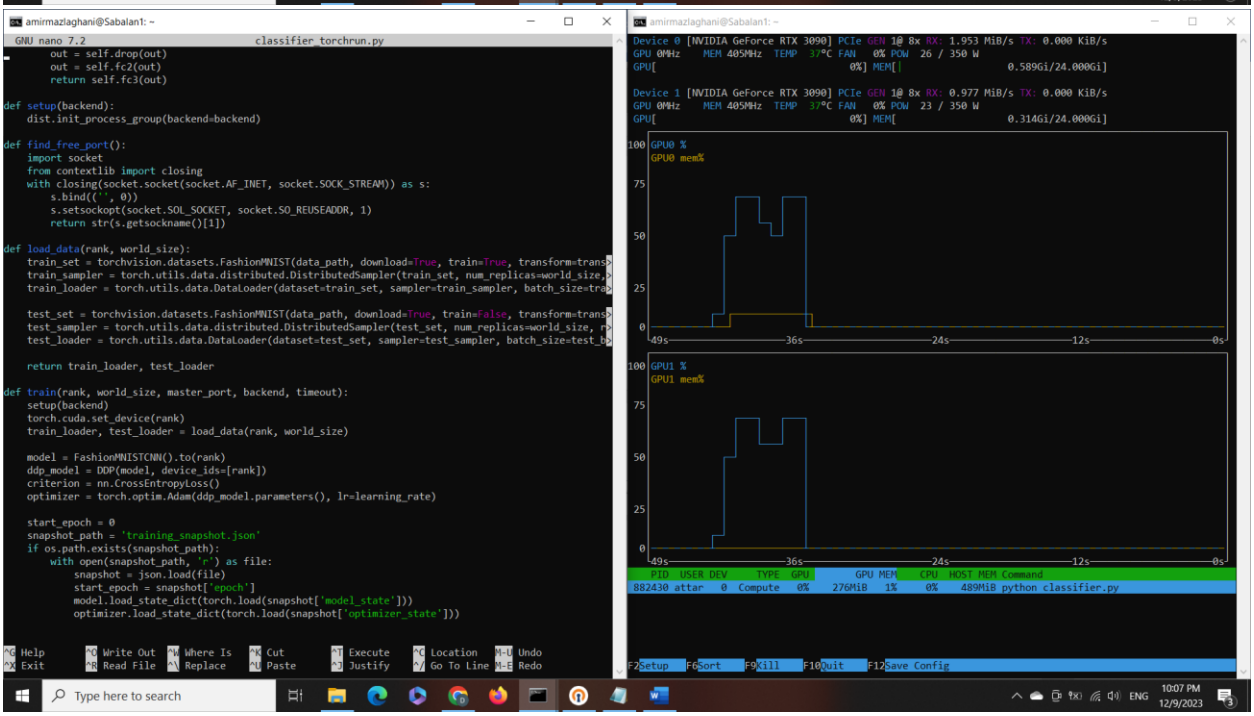

amirmazlaghani@Sabalan1: ~
GNU nano 7.2 classifier_mp.py
optimizer = torch.optim.Adam(ddp_model.parameters(), lr=learning_rate)
start_time = time.time()
for epoch in range(num_epochs):
model.train()
for i, (images, labels) in enumerate(train_loader):
images, labels = images.to(rank), labels.to(rank)
optimizer.zero_grad()
outputs = ddp_model(images)
loss = criterion(outputs, labels)
loss.backward()
optimizer.step()
if i % 100 == 99:
print(f"Rank: {rank}, Epoch: {epoch + 1}, Batch: {i + 1}, Loss: {loss.item():.3f}")
model.eval()
total = 0
correct = 0
with torch.no_grad():
for images, labels in test_loader:
images, labels = images.to(rank), labels.to(rank)
outputs = model(images)
_, predicted = torch.max(outputs.data, 1)
total += labels.size(0)
correct += (predicted == labels).sum().item()
accuracy = 100 * correct / total
print(f"Rank: {rank}, Epoch: {epoch + 1}, Accuracy: {accuracy:.2f}%")
end_time = time.time()
cuda_mem = torch.cuda.max_memory_allocated(device=rank) / (1024 ** 2)
print(f"Rank: {rank}, Training Time: {end_time - start_time} seconds, Max Memory Allocated: {cuda_mem} MB")
if rank == 0:
print("Finished Training")
if __name__ == "__main__":
world_size = torch.cuda.device_count()
master_port = find_free_port()
backend = 'nccl'
timeout = datetime.timedelta(seconds=10)
mp.spawn(train, nprocs=world_size, args=(world_size, master_port, backend, timeout), join=True)
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy
amirmazlaghani@Sabalan1: ~
Device 0 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000
GPU 0MHz MEM 405MHz TEMP 44°C FAN 0% POW 26 / 350 W
GPU[0%] MEM[0.313Gi/24.0
Device 1 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000
GPU 0MHz MEM 405MHz TEMP 38°C FAN 0% POW 22 / 350 W
GPU[0%] MEM[0.311Gi/24.0
GPU0 %
GPU0 mem%
GPU1 %
GPU1 mem%
36s 27s 18s 9s 0s
PID USER DEV TYPE
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.

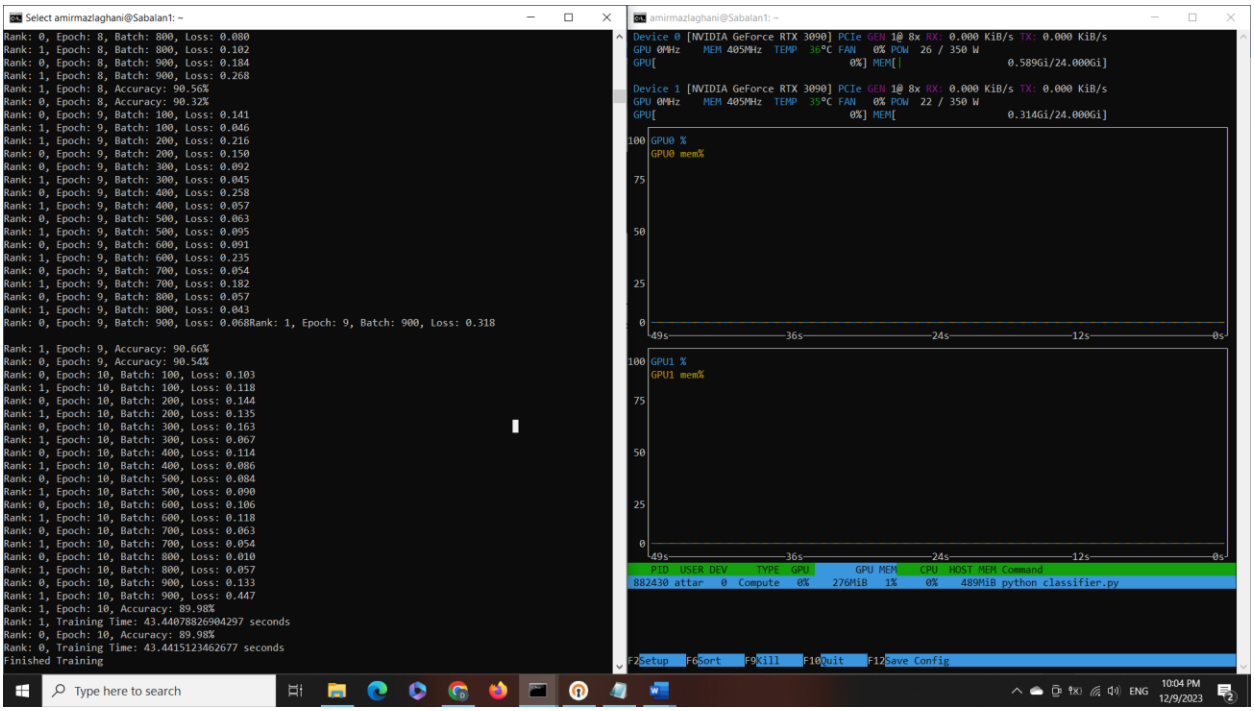


ج)

طبق ویدیویی که کمک‌مدرس در اختیارمان گذاشت، یک سری تغییرات شامل حذف متغیرهای محیطی و هندل شدن آن ها توسط torchrun صورت گرفت همچنین اضافه شدن اسنپ شات برای fault tolerance و تغییراتی دیگر در همین موارد. در ادامه کد و سپس خروجی آن را مشاهده میکنیم که شبیه به قسمت دو شد

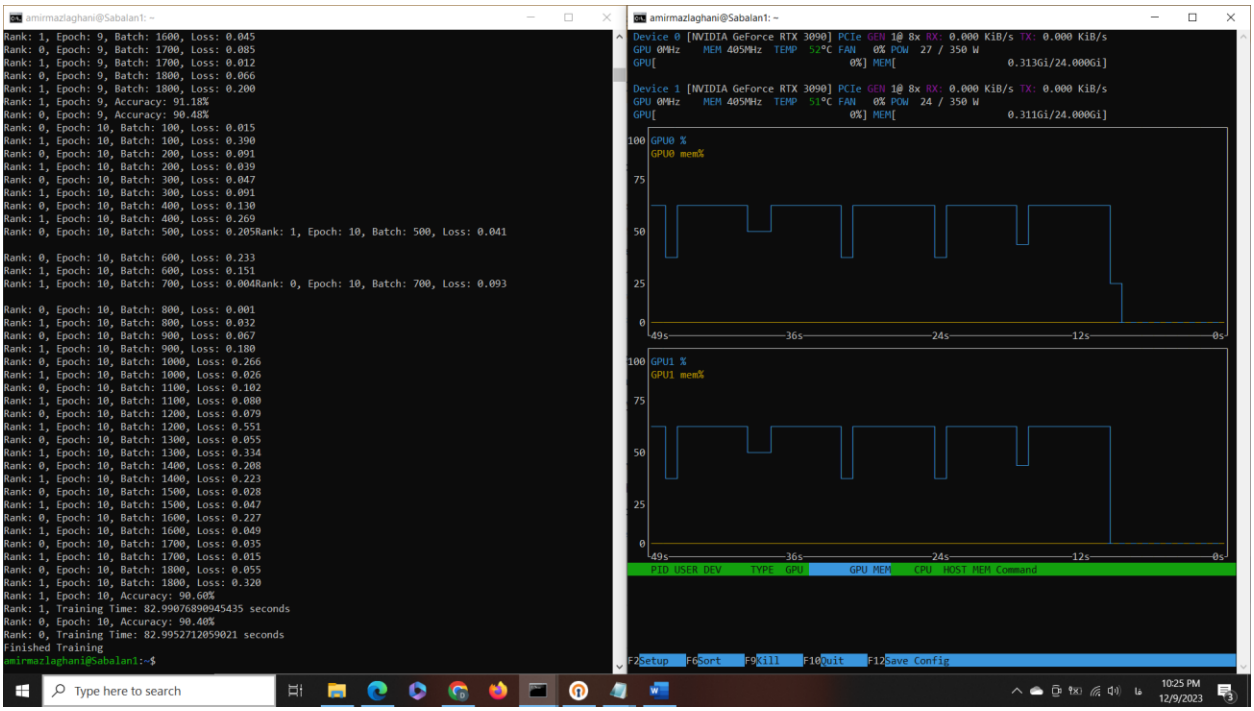

amirmazlaghani@Sabalan1: ~
GNU nano 7.2 classifier_torchrun.py
images, labels = images.to(rank), labels.to(rank)
optimizer.zero_grad()
outputs = ddp_model(images)
loss = criterion(outputs, labels)
loss.backward()
optimizer.step()
if i % 100 == 99:
print(f"Rank: {rank}, Epoch: {epoch + 1}, Batch: {i + 1}, Loss: {loss.item():.3f}")
model.eval()
total = 0
correct = 0
with torch.no_grad():
for images, labels in test_loader:
images, labels = images.to(rank), labels.to(rank)
outputs = model(images)
_, predicted = torch.max(outputs.data, 1)
total += labels.size(0)
correct += (predicted == labels).sum().item()
accuracy = 100 * correct / total
print(f"Rank: {rank}, Epoch: {epoch + 1}, Accuracy: {accuracy:.2f}%")
end_time = time.time()
print(f"Rank: {rank}, Training Time: {end_time - start_time} seconds")
if rank == 0:
torch.save(model.state_dict(), 'model_final.pt')
torch.save(optimizer.state_dict(), 'optimizer_final.pt')
with open('final_snapshot.json', 'w') as file:
json.dump({'epoch': num_epochs, 'model_state': 'model_final.pt', 'optimizer_state': 'optim
if rank == 0:
print("Finished Training")
if __name__ == "__main__":
rank = int(os.environ.get("LOCAL_RANK", 0))
world_size = torch.cuda.device_count()
master_port = find_free_port()
backend = 'nccl'
timeout = datetime.timedelta(seconds=10)
train(rank, world_size, master_port, backend, timeout)
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo
amirmazlaghani@Sabalan1: ~
Device 0 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000 KiB/s
GPU 0MHz MEM 405MHz TEMP 37°C FAN 0% POW 26 / 350 W
GPU[0%] MEM[0.589Gi/24.000Gi]
Device 1 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000 KiB/s
GPU 0MHz MEM 405MHz TEMP 37°C FAN 0% POW 22 / 350 W
GPU[0%] MEM[0.314Gi/24.000Gi]
GPU0 %
GPU0 mem%
GPU1 %
GPU1 mem%
49s 36s 24s 12s 0s
PID USER DEV TYPE GPU GPU MEM CPU HOST MEM Command
882430 attar 0 Compute 0% 276MiB 1% 0% 489MiB python classifier.py
F2Setup F6Sort F9Kill F10Quit F12Save Config
Type here to search
ENG
10:07 PM
12/9/2023


amirmazlaghani@Sabalan1: ~
GNU nano 7.2 classifier_torchrun.py
out = self.drop(out)
out = self.fc2(out)
return self.fc3(out)
def setup(backend):
dist.init_process_group(backend=backend)
def find_free_port():
import socket
from contextlib import closing
with closing(socket.socket(socket.AF_INET, socket.SOCK_STREAM)) as s:
s.bind(('', 0))
s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
return str(s.getsockname()[1])
def load_data(rank, world_size):
train_set = torchvision.datasets.FashionMNIST(data_path, download=True, train=True, transform=trans
train_sampler = torch.utils.data.distributed.DistributedSampler(train_set, num_replicas=world_size,
train_loader = torch.utils.data.DataLoader(dataset=train_set, sampler=train_sampler, batch_size=tra
test_set = torchvision.datasets.FashionMNIST(data_path, download=True, train=False, transform=trans
test_sampler = torch.utils.data.distributed.DistributedSampler(test_set, num_replicas=world_size, r
test_loader = torch.utils.data.DataLoader(dataset=test_set, sampler=test_sampler, batch_size=test_b
return train_loader, test_loader
def train(rank, world_size, master_port, backend, timeout):
setup(backend)
torch.cuda.set_device(rank)
train_loader, test_loader = load_data(rank, world_size)
model = FashionMNISTCNN().to(rank)
ddp_model = DDP(model, device_ids=[rank])
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(ddp_model.parameters(), lr=learning_rate)
start_epoch = 0
snapshot_path = 'training_snapshot.json'
if os.path.exists(snapshot_path):
with open(snapshot_path, 'r') as file:
snapshot = json.load(file)
start_epoch = snapshot['epoch']
model.load_state_dict(torch.load(snapshot['model_state']))
optimizer.load_state_dict(torch.load(snapshot['optimizer_state']))
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo
amirmazlaghani@Sabalan1: ~
Device 0 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 1.953 MiB/s TX: 0.000 KiB/s
GPU 0MHz MEM 405MHz TEMP 37°C FAN 0% POW 26 / 350 W
GPU[0%] MEM[0.589Gi/24.000Gi]
Device 1 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.977 MiB/s TX: 0.000 KiB/s
GPU 0MHz MEM 405MHz TEMP 37°C FAN 0% POW 23 / 350 W
GPU[0%] MEM[0.314Gi/24.000Gi]
GPU0 %
GPU0 mem%
GPU1 %
GPU1 mem%
49s 36s 24s 12s 0s
PID USER DEV TYPE GPU GPU MEM CPU HOST MEM Command
882430 attar 0 Compute 0% 276MiB 1% 0% 489MiB python classifier.py
F2Setup F6Sort F9Kill F10Quit F12Save Config
Type here to search
ENG
10:07 PM
12/9/2023

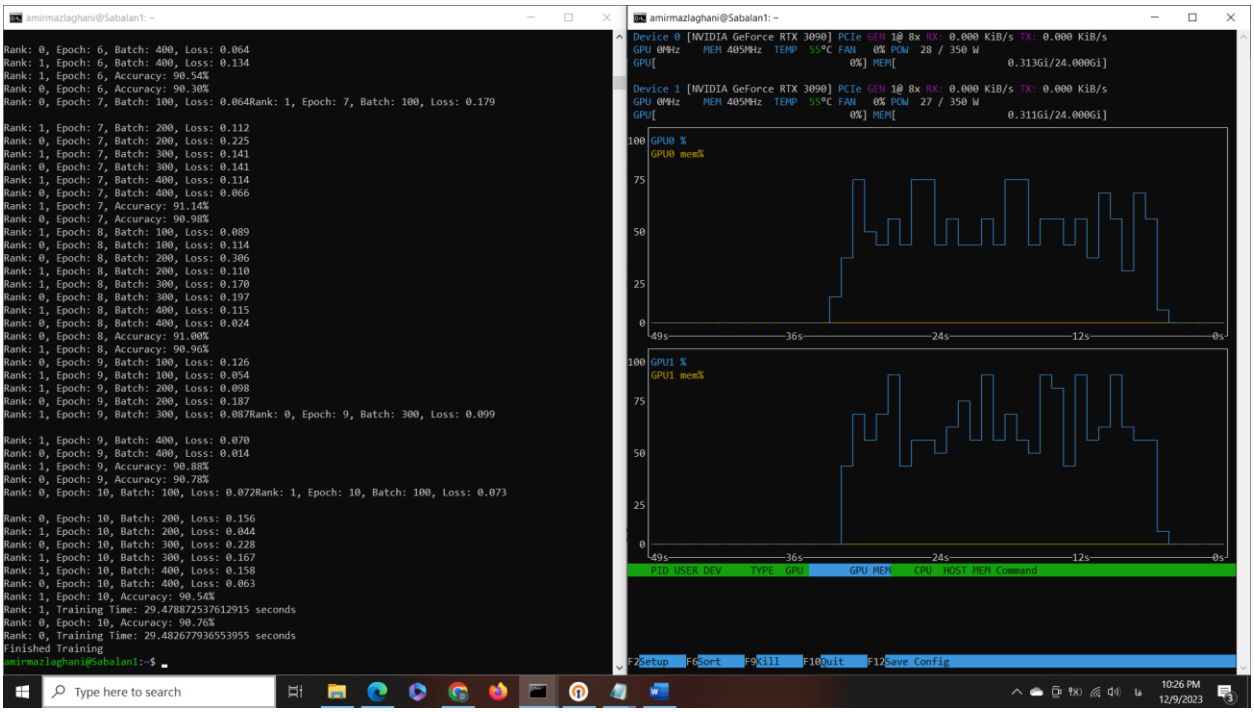


سوال دوم:

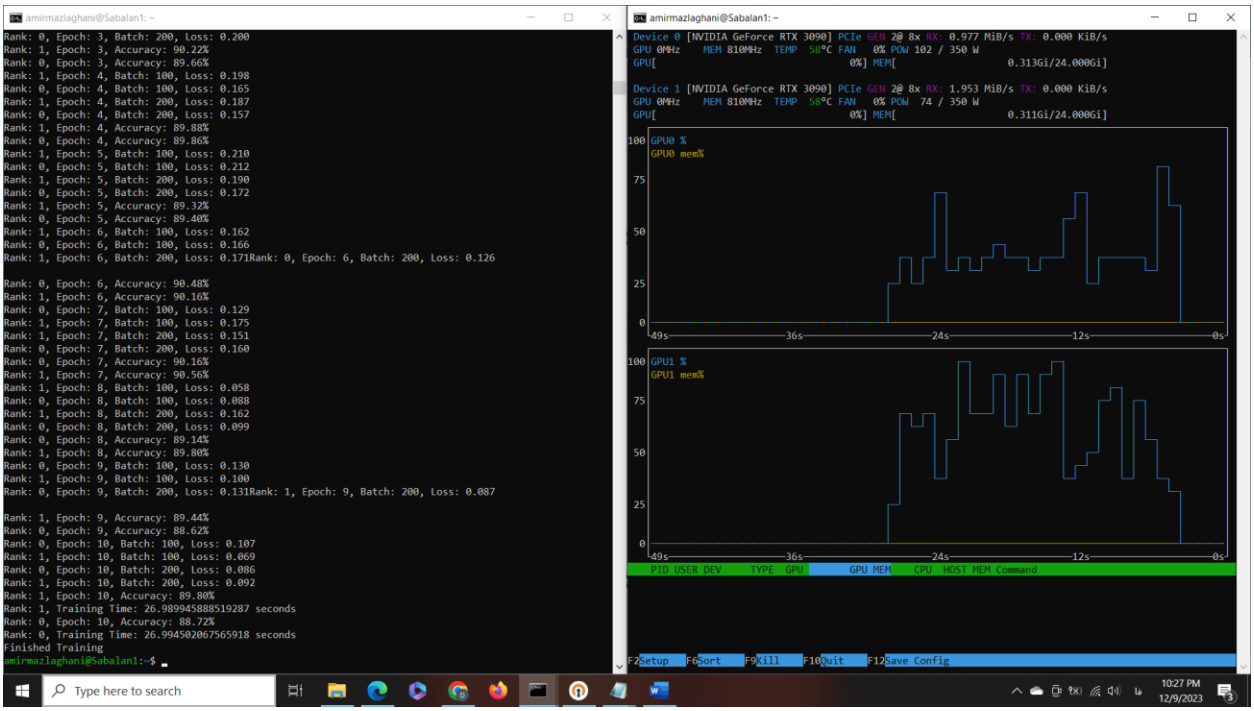
سایز ۱۶ :



سایز ۶۴:



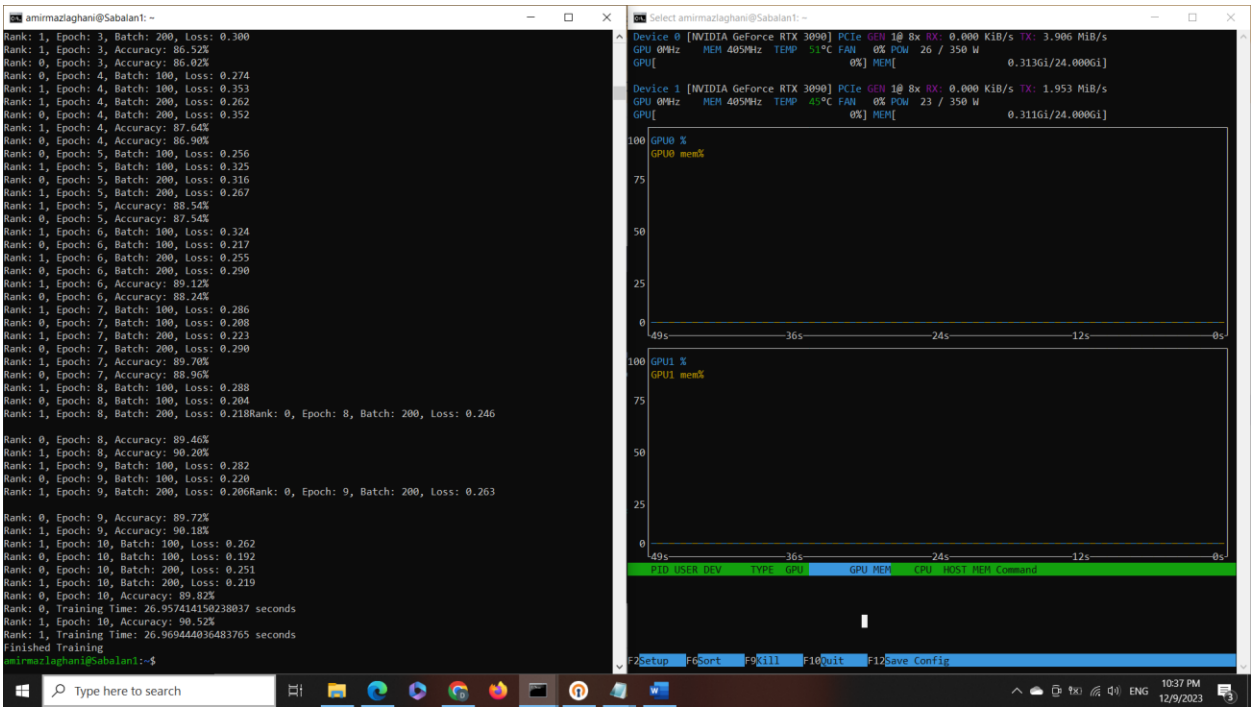
سایز ۱۲۸:



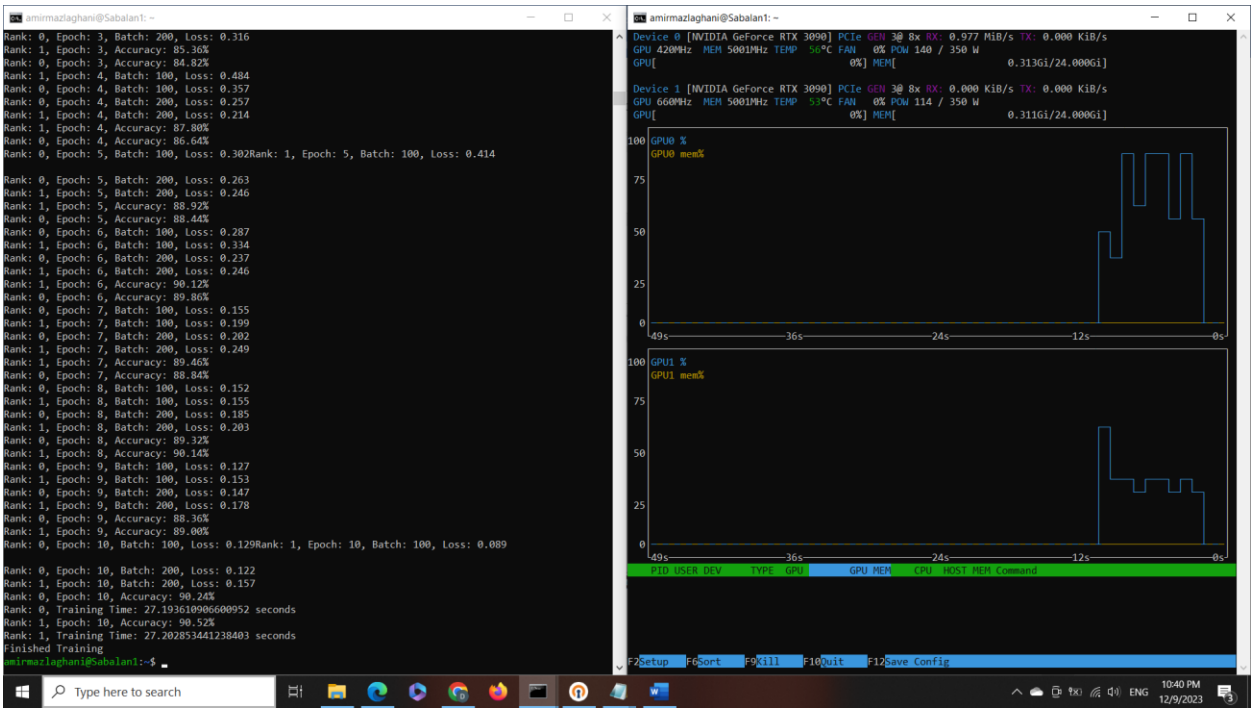
تغییرات per epoch هم در لاگ تصاویر موجود است.(وقت برای نمودار کشیدن نبود 😊)

سوال چهار :

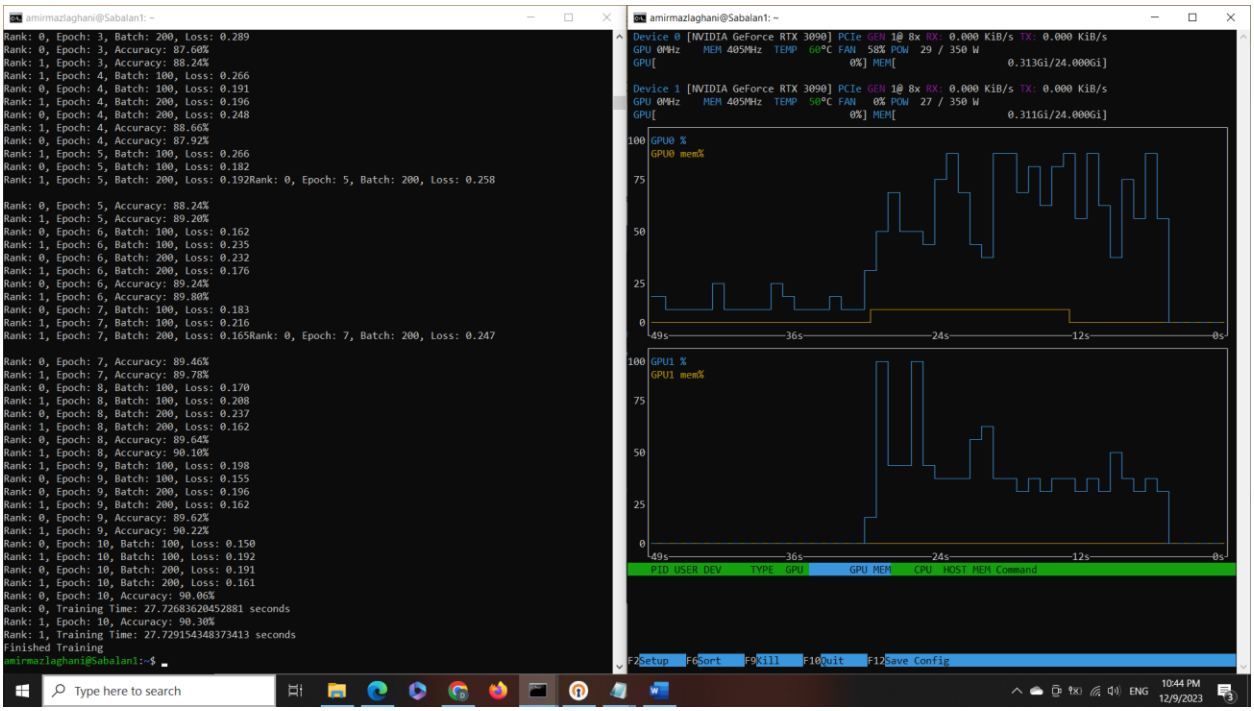
SGD:کد در ضمیمه موجود است



:RMSProp



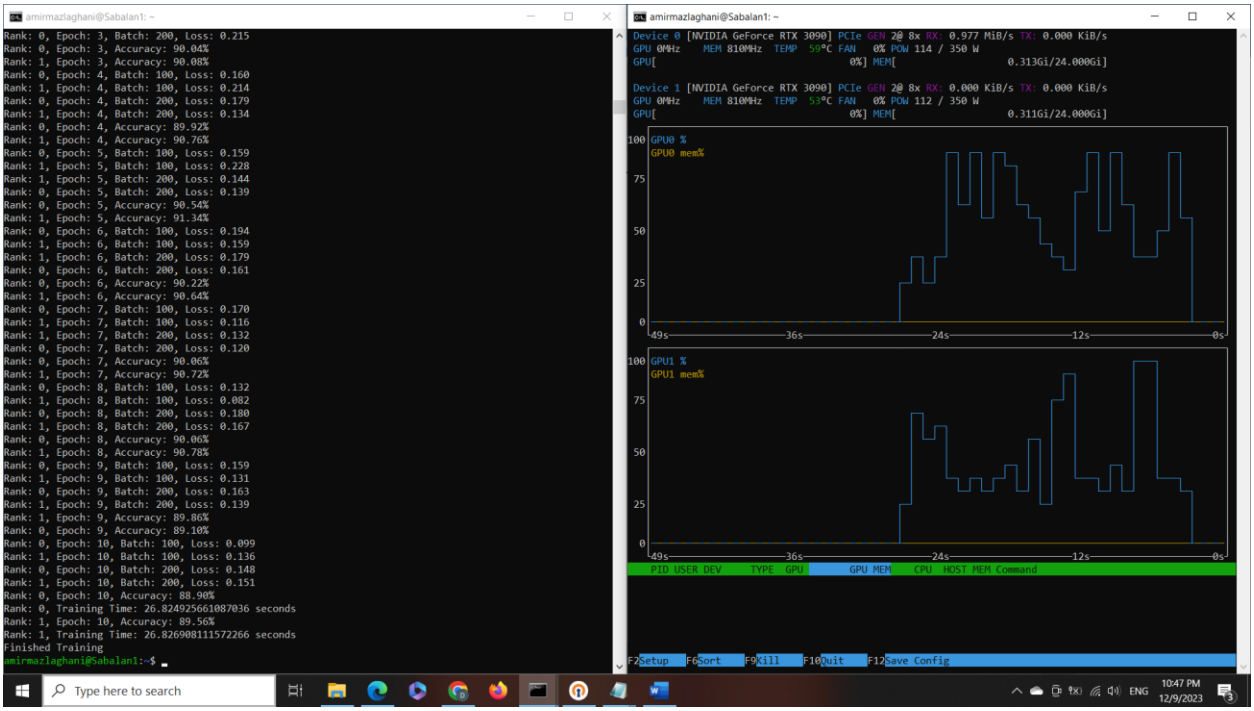
:Adagrad



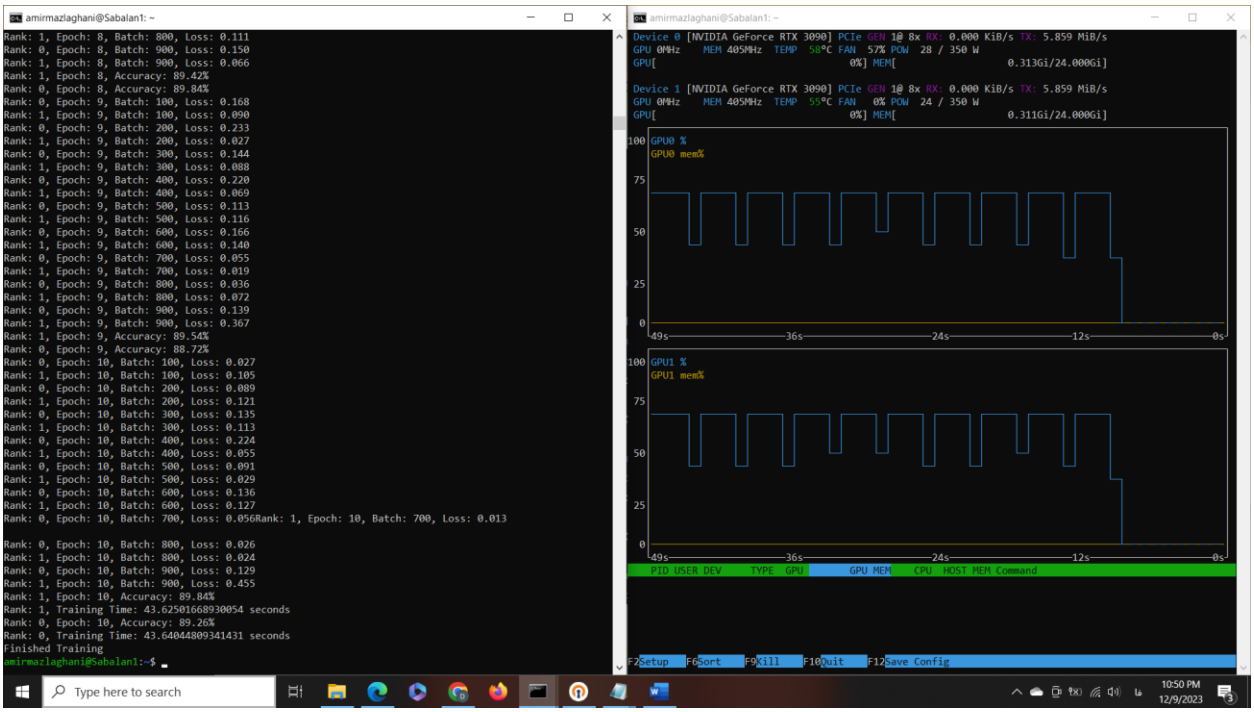
کد تمامی قسمت ها به صورت تفکیک شده در ضمیمه موجود است.

سوال پنج:

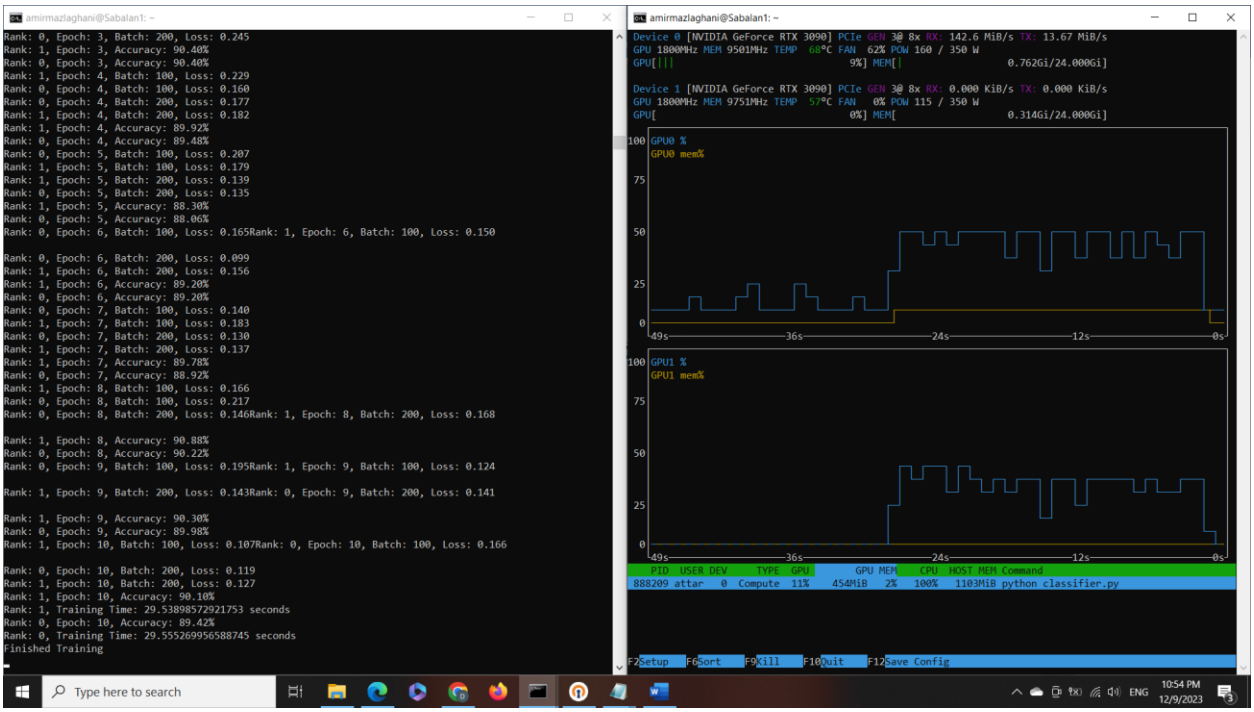
Nccl 128:



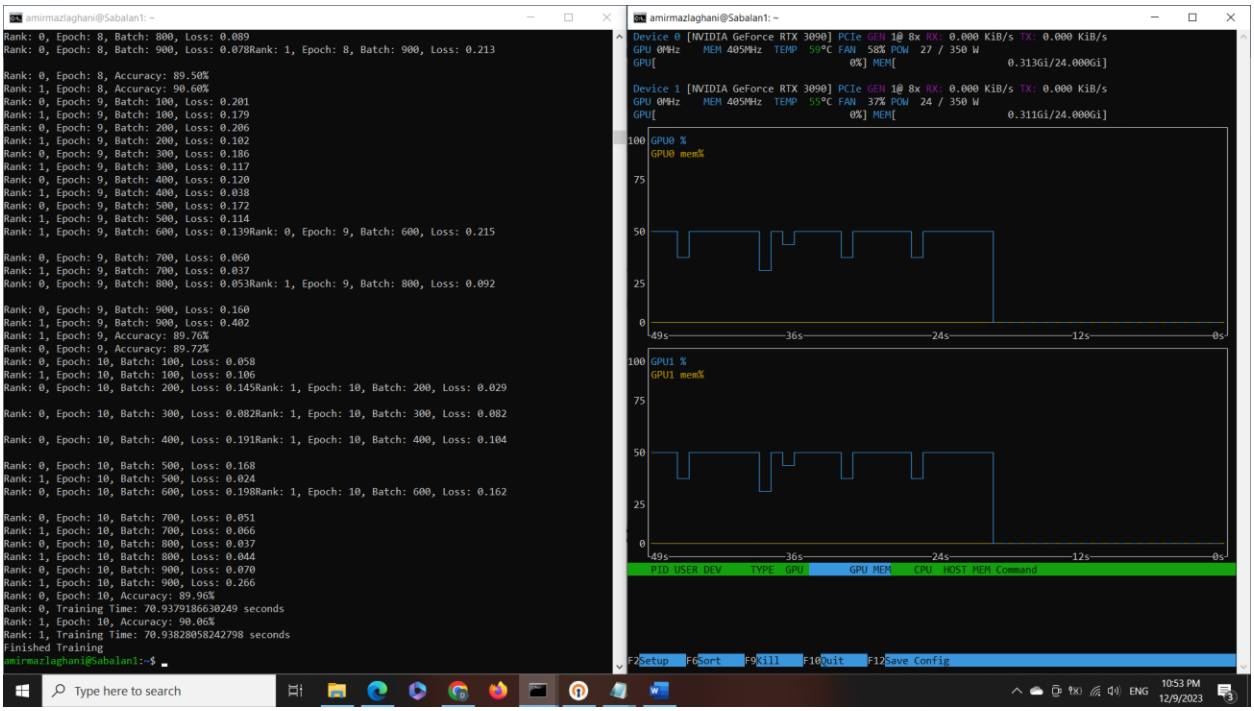
Nccl 32 :



:Gloo 128



:Gloo 32


amirmazlaghani@Sabalan1: ~
Rank: 0, Epoch: 8, Batch: 800, Loss: 0.089
Rank: 0, Epoch: 8, Batch: 900, Loss: 0.078Rank: 1, Epoch: 8, Batch: 900, Loss: 0.213
Rank: 0, Epoch: 8, Accuracy: 89.50%
Rank: 1, Epoch: 8, Accuracy: 90.60%
Rank: 0, Epoch: 9, Batch: 100, Loss: 0.201
Rank: 1, Epoch: 9, Batch: 100, Loss: 0.179
Rank: 0, Epoch: 9, Batch: 200, Loss: 0.206
Rank: 1, Epoch: 9, Batch: 200, Loss: 0.102
Rank: 0, Epoch: 9, Batch: 300, Loss: 0.186
Rank: 1, Epoch: 9, Batch: 300, Loss: 0.117
Rank: 0, Epoch: 9, Batch: 400, Loss: 0.120
Rank: 1, Epoch: 9, Batch: 400, Loss: 0.038
Rank: 0, Epoch: 9, Batch: 500, Loss: 0.172
Rank: 1, Epoch: 9, Batch: 500, Loss: 0.114
Rank: 1, Epoch: 9, Batch: 600, Loss: 0.139Rank: 0, Epoch: 9, Batch: 600, Loss: 0.215
Rank: 0, Epoch: 9, Batch: 700, Loss: 0.060
Rank: 1, Epoch: 9, Batch: 700, Loss: 0.037
Rank: 0, Epoch: 9, Batch: 800, Loss: 0.053Rank: 1, Epoch: 9, Batch: 800, Loss: 0.092
Rank: 0, Epoch: 9, Batch: 900, Loss: 0.160
Rank: 1, Epoch: 9, Batch: 900, Loss: 0.402
Rank: 1, Epoch: 9, Accuracy: 89.76%
Rank: 0, Epoch: 9, Accuracy: 89.72%
Rank: 0, Epoch: 10, Batch: 100, Loss: 0.058
Rank: 1, Epoch: 10, Batch: 100, Loss: 0.106
Rank: 0, Epoch: 10, Batch: 200, Loss: 0.145Rank: 1, Epoch: 10, Batch: 200, Loss: 0.029
Rank: 0, Epoch: 10, Batch: 300, Loss: 0.082Rank: 1, Epoch: 10, Batch: 300, Loss: 0.082
Rank: 0, Epoch: 10, Batch: 400, Loss: 0.191Rank: 1, Epoch: 10, Batch: 400, Loss: 0.104
Rank: 0, Epoch: 10, Batch: 500, Loss: 0.168
Rank: 1, Epoch: 10, Batch: 500, Loss: 0.024
Rank: 0, Epoch: 10, Batch: 600, Loss: 0.198Rank: 1, Epoch: 10, Batch: 600, Loss: 0.162
Rank: 0, Epoch: 10, Batch: 700, Loss: 0.051
Rank: 1, Epoch: 10, Batch: 700, Loss: 0.066
Rank: 0, Epoch: 10, Batch: 800, Loss: 0.037
Rank: 1, Epoch: 10, Batch: 800, Loss: 0.044
Rank: 0, Epoch: 10, Batch: 900, Loss: 0.070
Rank: 1, Epoch: 10, Batch: 900, Loss: 0.266
Rank: 0, Epoch: 10, Accuracy: 89.96%
Rank: 0, Training Time: 70.9379186630249 seconds
Rank: 1, Epoch: 10, Accuracy: 90.06%
Rank: 1, Training Time: 70.93828058242798 seconds
Finished Training
amirmazlaghani@Sabalan1:~$
amirmazlaghani@Sabalan1: ~
Device 0 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000 KiB/s
GPU 0MHz MEM 405MHz TEMP 59°C FAN 58% POW 27 / 350 W
GPU[0%] MEM[0.313Gi/24.000Gi]
Device 1 [NVIDIA GeForce RTX 3090] PCIe GEN 1@ 8x RX: 0.000 KiB/s TX: 0.000 KiB/s
GPU 0MHz MEM 405MHz TEMP 55°C FAN 37% POW 24 / 350 W
GPU[0%] MEM[0.311Gi/24.000Gi]
100
75
50
25
0
GPU0 %
GPU0 mem%
49s 36s 24s 12s 0s
100
75
50
25
0
GPU1 %
GPU1 mem%
49s 36s 24s 12s 0s
PID USER DEV TYPE GPU GPU MEM CPU HOST MEM Command
F2Setup F6Sort F9Kill F10Quit F12Save Config
Type here to search
ENG
10:53 PM
12/9/2023