

Instructions

1. You have 2 hour to solve this quiz.
2. Please make sure you have proper Internet connection and a laptop before you proceed.
3. This is a coding exercise.
4. Even if you are working in a group, separate submissions are required, and you cannot just name your group member.

This quiz was locked May 31 at 6am.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	113 minutes	0 out of 1,000 *
* Some questions not yet graded			

Score for this quiz: 0 out of 1,000 *
Submitted May 30 at 5:04pm
This attempt took 113 minutes.

Question 1

Not yet graded / 1,000 pts

Make a DNN such that:

1. It's first block uses following code:

1. import datetime from datetime

2. print("Current Date/Time: ", datetime.now())

2. uses the Modules you have written

3. calls following DNN from a file called QuizDNN.py:

1. x1 = Input

2. x2 = Conv(x1)

3. x3 = Conv(x1 + x2)

4. x4 = MaxPooling(x1 + x2 + x3)

5. x5 = Conv(x4)

6. x6 = Conv(x4 + x5)

7. x7 = Conv(x4 + x5 + x6)

8. x8 = MaxPooling(x5 + x6 + x7)

9. x9 = Conv(x8)

10. x10 = Conv (x8 + x9)

11. x11 = Conv (x8 + x9 + x10)

12. x12 = GAP(x11)

13. x13 = FC(x12)

4. Uses ReLU and BN wherever applicable

5. Uses CIFAR10 as the dataset

6. Your target is 75% in less than 40 Epochs

7. Once Done:

1. Paste the code in QuizDNN.py

2. Paste the code of your Colab (or computer's) Notebook

3. Paste the complete training log

4. Paste the link to your Google Colab Notebook file (or you GitHub Repo)

If there is timing mismatch in any of the files confirming that you attempted the quiz outside the window, then you'll get -50% of this quiz marks.

Your Answer:

Colab's Notebook:

""S9_QuizDNN.ipynb

Automatically generated by Colaboratory.

Original file is located at
https://colab.research.google.com/drive/1R3QyZlijXn0UzxfLeQGEkq2MAI_cYKSW

#Module Used - data.py train.py, test.py, Albtransform.py, summary2.py, utils.py,classifier.py

""

#Time

from datetime import datetime
print("Current Date/Time: ", datetime.now())

#Getting Data

import Albtransform

import data
train_loader, test_loader = data.loader(128)
data.display()

#Getting Model

import QuizDNN as m

!pip install torchsummary
import summary2 as summary
summary.summ(device)

#Importing Train and Test Module

import train as tr
import test as te

#Starting Training

import torch.optim as optim
from torch.optim.lr_scheduler import StepLR

model = m.Net().to(device)

optimizer = optim.SGD(model.parameters(), lr=0.01, momentum=0.9, weight_decay=5e-4)

epochs = 10

for epoch in range(epochs):
 print('EPOCH: ', epoch+1)
 tr.train(model, device, train_loader, optimizer, epoch)
 te.test(model, device, test_loader)

#Visualisation

import utils as ut

path = '/content/classifier.pt'
saved_model = m.Net().to(device)
saved_model.load_state_dict(torch.load(path))
saved_model.eval()
ut.mis(saved_model, device, test_loader, 25)

ut.graph()

ut.testvtrain()

ut.class_acc(saved_model,device, test_loader)

#Training Log

0%| | 0/391 [00:00<?, ?it/s]

EPOCH: 1

Loss=1.2398136854171753 Batch_id=390 Accuracy=48.71: 100%| | 391/391 [00:24<00:00, 15.791t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0095, Accuracy: 5658/10000 (56.58%)

EPOCH: 2

Loss=1.0150179862976074 Batch_id=390 Accuracy=64.63: 100%| | 391/391 [00:26<00:00, 14.951t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0077, Accuracy: 6466/10000 (64.66%)

EPOCH: 3

Loss=0.7135049104690552 Batch_id=390 Accuracy=71.31: 100%| | 391/391 [00:25<00:00, 15.391t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0061, Accuracy: 7282/10000 (72.82%)

EPOCH: 4

Loss=0.5388146042823792 Batch_id=390 Accuracy=75.57: 100%| | 391/391 [00:25<00:00, 15.561t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0054, Accuracy: 7523/10000 (75.23%)

EPOCH: 5

Loss=0.5179477334022522 Batch_id=390 Accuracy=78.54: 100%| | 391/391 [00:25<00:00, 15.221t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0053, Accuracy: 7657/10000 (76.57%)

EPOCH: 6

Loss=0.45289012488365173 Batch_id=390 Accuracy=80.23: 100%| | 391/391 [00:25<00:00, 15.321t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0052, Accuracy: 7707/10000 (77.07%)

EPOCH: 7

Loss=0.6214013695716858 Batch_id=390 Accuracy=82.06: 100%| | 391/391 [00:25<00:00, 15.371t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0046, Accuracy: 7994/10000 (79.94%)

EPOCH: 8

Loss=0.46650147438049316 Batch_id=390 Accuracy=82.92: 100%| | 391/391 [00:25<00:00, 15.291t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0045, Accuracy: 7997/10000 (79.97%)

EPOCH: 9

Loss=0.3674827218055725 Batch_id=390 Accuracy=84.07: 100%| | 391/391 [00:25<00:00, 15.291t/s]
0%| | 0/391 [00:00<?, ?it/s]

Test set: Average loss: 0.0045, Accuracy: 8082/10000 (80.82%)

EPOCH: 10

Loss=0.4949423670768738 Batch_id=390 Accuracy=84.75: 100%| | 391/391 [00:25<00:00, 15.291t/s]

Test set: Average loss: 0.0042, Accuracy: 8155/10000 (81.55%)

#Colab Link

https://colab.research.google.com/drive/1R3QyZlijXn0UzxfLeQGEkq2MAI_cYKSW?usp=sharing ↗

#GitHub Link

https://github.com/abksyed/EVA4/blob/master/O9_DataAugmentation/QuizDNN/S9_QuizDNN.ipynb ↗

#QuizDNN.py

https://github.com/abksyed/EVA4/blob/master/O9_DataAugmentation/QuizDNN/QuizDNN.py ↗

#Readme File

https://github.com/abksyed/EVA4/blob/master/O9_DataAugmentation/QuizDNN/Readme.md