Machine Learning & Deep Learning (Barcha uchun)

<08> PyTorch DataLoader

Mansurbek Abdullaev


https://uzbek.gitbook.io/ai/


mansurbek.comchemai@gmail.com


@MansurbekUST

Ma’lumotlarni manual tarzda tizimga yuklash (Manual data feed)

```python
#Ma'lumotlarni numpy yordamida yuklab olish
xy_data = np.loadtxt('../Data/diabetes.csv', delimiter=',', dtype = np.float32)
# x va y data larga ajratib chiqish (Traning data)
x_data = torch.from_numpy(xy_data[:750, 0:-1])
y_data = torch.from_numpy(xy_data[:750, [-1]])
```

```python
# Training
for epoch in range(15000):      # Epochlar soni 15000
    y_pred = model(x_data)  # Forward (to'g'ri xisoblash)
    loss = criterioin(y_pred, y_data)
    if epoch % 1000 == 0:# loss ni hisoblash
        print(f'epoch # {epoch} --> Loss {loss.item():.4f}')
    optimizer.zero_grad()  # Optimizerni nolga tenglab olish
    loss.backward() # Teskari hisoblash (Back prop)
    optimizer.step()  # Optimizer orqali w ni qiymatini yangilash
```

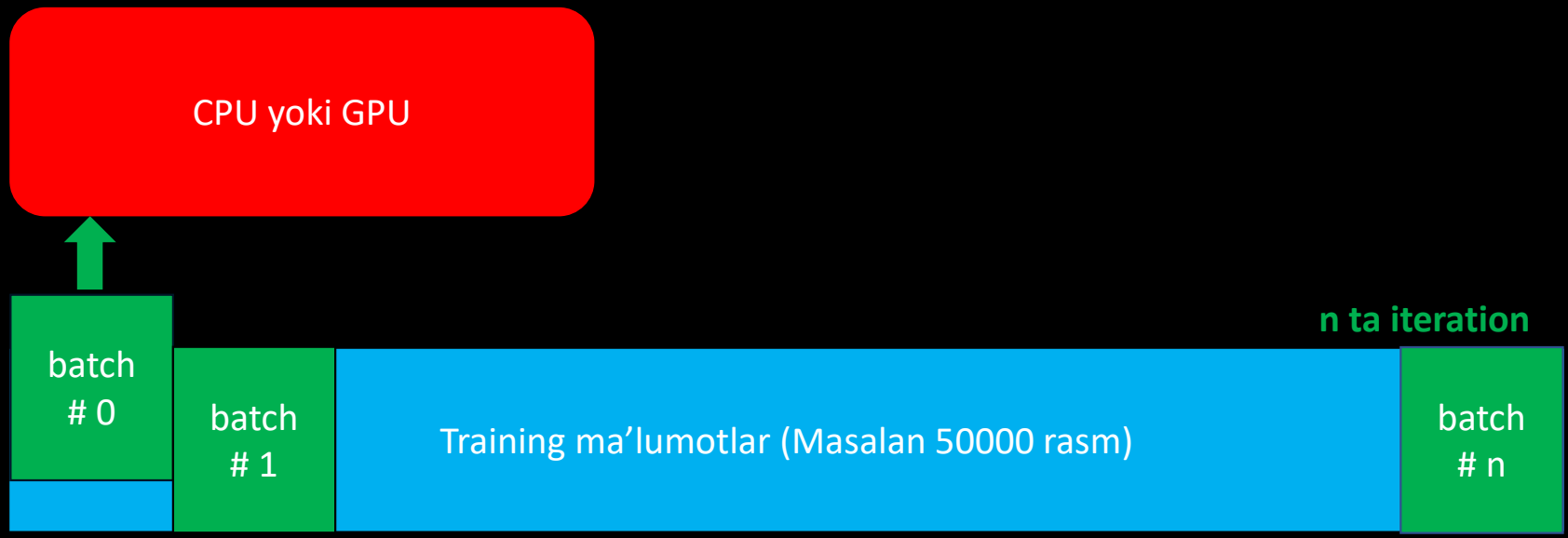
Batch

```
# Training sikl
for epoch in range(training_epochs):
    # Batch sikl
    for i in range(total_batch)
```

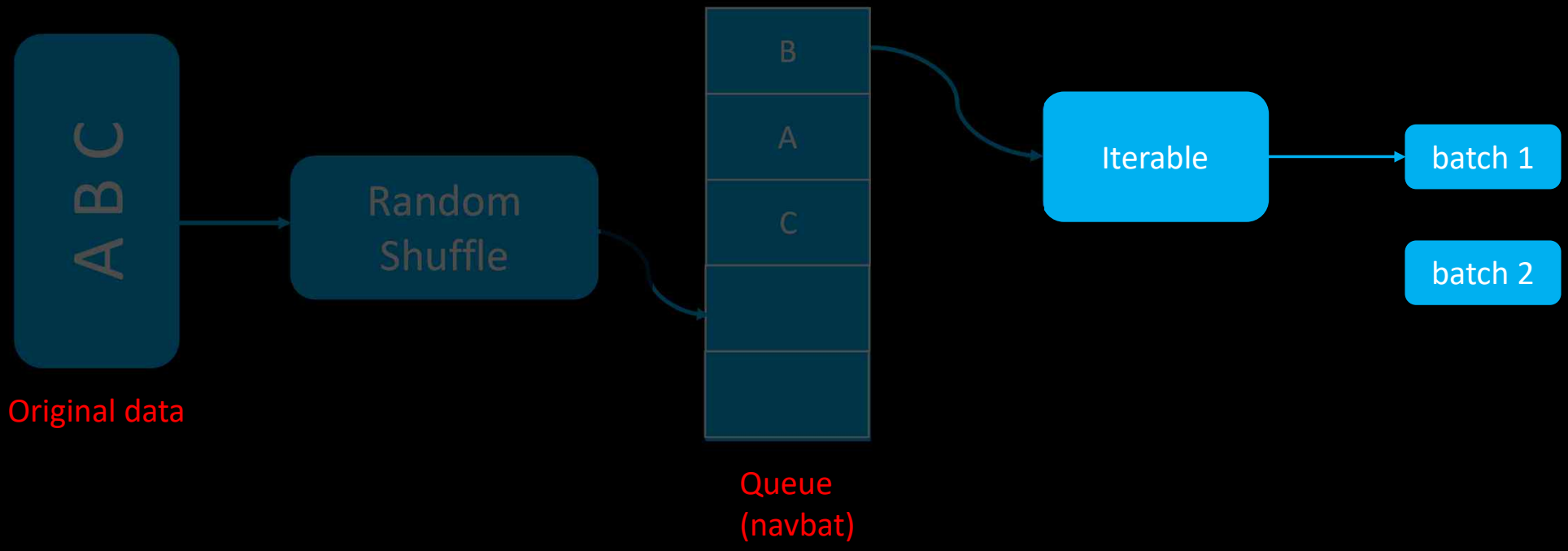
Neyron tarmoqlaridagi terminlar:

- Batch size → Har bitta batchdagi ma’lumotlar soni [Barcha ma’lumotlar/batchlar soni]
- Iteration → Har bir kichik batch ning bir marta hisoblanishi.
- Bir epoch → Barcha training ma’lumotlarni bir marta to’liq (forward & backward) hisoblab chiqish.

Nimaga DataLoader ?



DataLoader


A B C
Original data
Random Shuffle
B
A
C
Queue (navbat)
Iterable
batch 1
batch 2

Ixtiyoriy(Custom) DataLoader

```python
from torch.utils.data import Dataset, DataLoader

class CustomDataset(Dataset):

    # Ma'lumotlarni tayyorlash
    def __init__(self):


    def __getitem__(self, index):
        return


    def __len__(self):
        return

dataset = CustomDataset()
train_loader = DataLoader(dataset=dataset,
                          batch_size=64,
                          shuffle=True)
```

1. Yuklash, o’qish, tensorga o’girish va h.k.
2. Har bir indeksga ko’ra har bir elementni olish
3. Ma’lumotlarning uzunligini olish

DataLoaderga misol

```python
# Ma'lumotlarni ham class yordamida tartibga keltirib DataLoaderdan foydalanamiz.
class DiabetesDataset(Dataset):
    """ Qandli diabet ma'lumotlar to'plami"""

    # Ma'lumotlarga dastalbki ishlov berish, yuklab olish, tensorga o'girish
    def __init__(self):
        xy = np.loadtxt('../Data/diabetes.csv.',
                        delimiter=',', dtype=np.float32)
        self.len = xy.shape[0]
        self.x_data = torch.from_numpy(xy[:, 0:-1])
        self.y_data = torch.from_numpy(xy[:, [-1]])

    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

    def __len__(self):
        return self.len


dataset = DiabetesDataset()
train_loader = DataLoader(dataset=dataset,
                          batch_size=64,
                          shuffle=True)
```

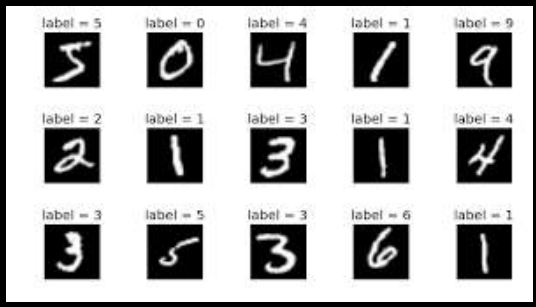
DataLoaderga misol

```python
# MNIST dataset
train_dataset = torchvision.datasets.MNIST(root='../../data',
                                           train=True,
                                           transform=transforms.ToTensor(),
                                           download=True)

test_dataset = torchvision.datasets.MNIST(root='../../data',
                                          train=False,
                                          transform=transforms.ToTensor())

# Data loader
train_loader = torch.utils.data.DataLoader(dataset=train_dataset,
                                           batch_size=batch_size,
                                           shuffle=True)

test_loader = torch.utils.data.DataLoader(dataset=test_dataset,
                                          batch_size=batch_size,
                                          shuffle=False)
```



https://github.com/keineahnung2345/pytorch-tutorial-jupyter-notebooks/

DataLoaderga ko’proq misol

- MNIST and FashionMNIST
- COCO (Captioning and Detection)
- LSUN Classification
- ImageFolder
- Imagenet-l2
- CIFAR10 and CIFAR100
- STL10
- SVHN
- PhotoTour

Keyingi dars

Softmax classification