神经网络

数据集

> 手写数据集MINST

```python
trainDataset = MNIST('../data/',train=True , transform=ToTensor() , dc
testDataset = MNIST('../data/',train=False , transform=ToTensor() , dc

print(len(trainDataset))
print(len(testDataset))

trainDataLoader = DataLoader(trainDataset , 32 , shuffle=True)
testDataLoader = DataLoader(testDataset , 32 , shuffle=True)
```

```python
import torch
from torch.utils.data import DataLoader
from torchvision.datasets import MNIST
from torchvision.transforms import ToTensor
from tensorboardX import SummaryWriter
from torch import nn
from torch.optim import SGD



writer = SummaryWriter('logs')

trainDataset = MNIST('../data/',train=True , transform=ToTensor() , dc
testDataset = MNIST('../data/',train=False , transform=ToTensor() , dc

print(len(trainDataset))
print(len(testDataset))

trainDataLoader = DataLoader(trainDataset , 32 , shuffle=True)
```

```
testDataLoader = DataLoader(testDataset , 32 , shuffle=True)
step = 0
net = nn.Sequential(
    nn.Flatten(),
    nn.Linear(784,256),
    nn.Linear(256,256),
    nn.Linear(256,10)
)

loss = nn.CrossEntropyLoss()
lr = 0.05
optim = SGD(net.parameters() ,lr )
epochs = 20

for epoch in range(epochs):
    net.train()
    trainloss = 0
 step = 0
 for imgs , targets in trainDataLoader:
        outs = net(imgs)
        l = loss(outs,targets)
        optim.zero_grad()
        l.backward()
        optim.step()
        trainloss = trainloss + l
        writer.add_images('trainData',imgs,step)
        step += 1
 trainloss = trainloss / step
    writer.add_scalar('trainloss', trainloss, epoch)
    # 验证
 net.eval()
    with torch.no_grad():
        step = 0
 testloss = 0
 for imgs , targets in trainDataLoader:
            outs = net(imgs)
            l = loss(outs, targets)
            testloss += l
```

```
              writer.add_images('testData',imgs,step)
              step += 1
 testloss = testloss / step
        writer.add_scalar('testloss',testloss,epoch)

writer.close()
```