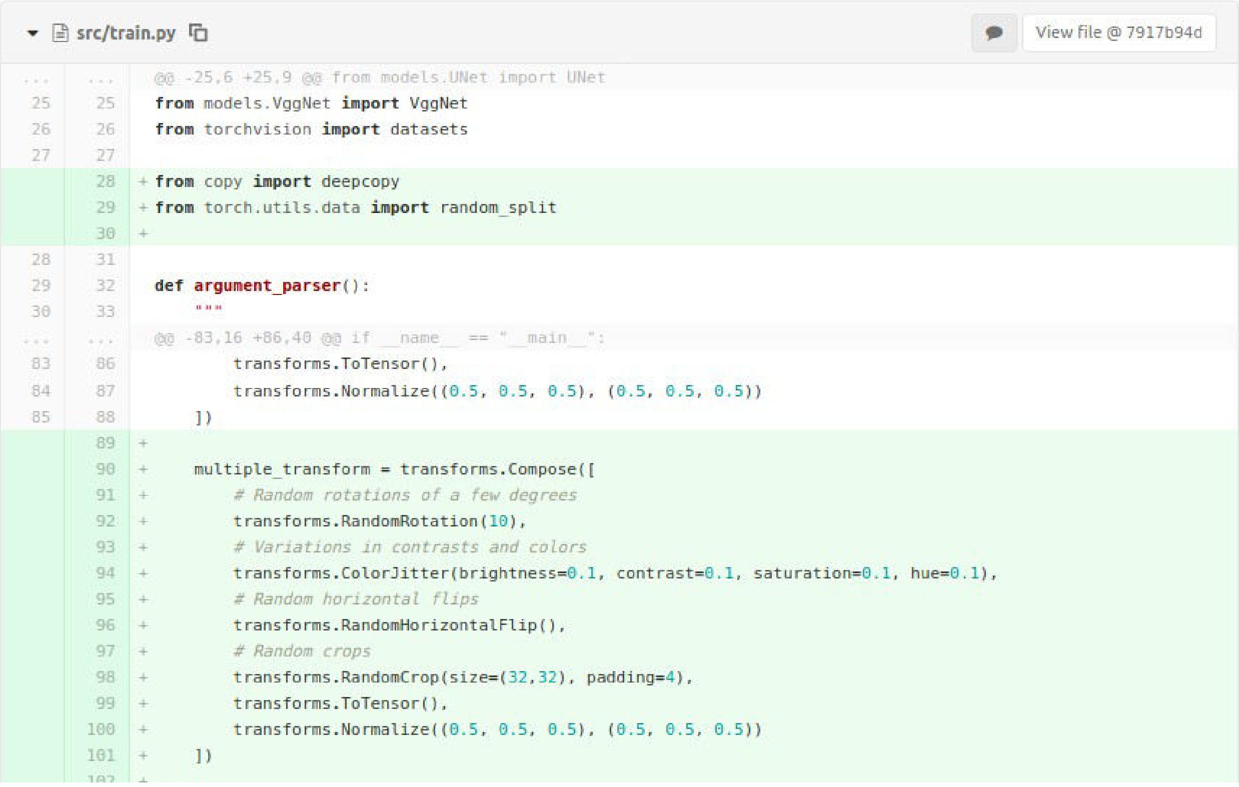
IFT 725 : Devoir 3

Question 2 :

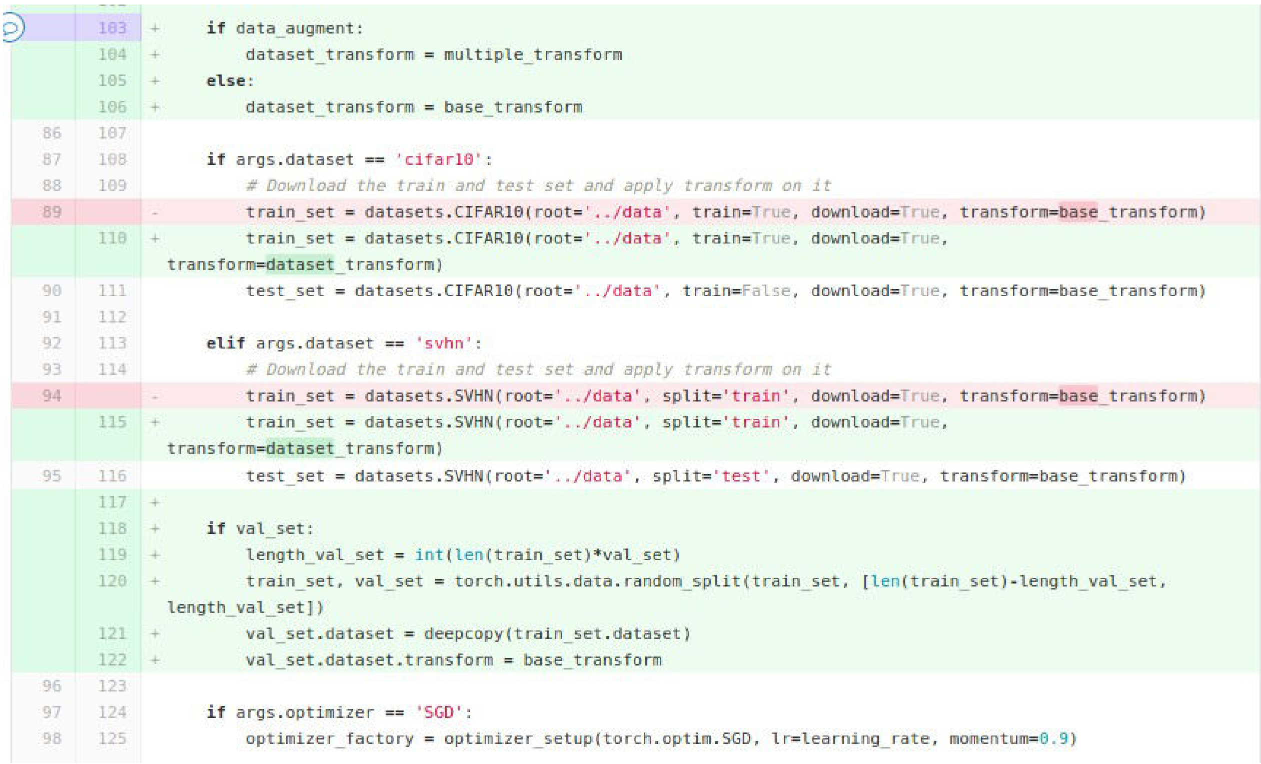
1. Endroits où du code a été ajoutée :



src/train.py

View file @ 7917b94d

```
@@ -25,6 +25,9 @@ from models.UNet import UNet
 from models.VggNet import VggNet
 from torchvision import datasets
 
+from copy import deepcopy
+from torch.utils.data import random_split
+
 
 def argument_parser():
     """
@@ -83,16 +86,40 @@ if __name__ == "__main__":
         transforms.ToTensor(),
         transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
     ])
+
+    multiple_transform = transforms.Compose([
+        # Random rotations of a few degrees
+        transforms.RandomRotation(10),
+        # Variations in contrasts and colors
+        transforms.ColorJitter(brightness=0.1, contrast=0.1, saturation=0.1, hue=0.1),
+        # Random horizontal flips
+        transforms.RandomHorizontalFlip(),
+        # Random crops
+        transforms.RandomCrop(size=(32,32), padding=4),
+        transforms.ToTensor(),
+        transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
+    ])
+
```



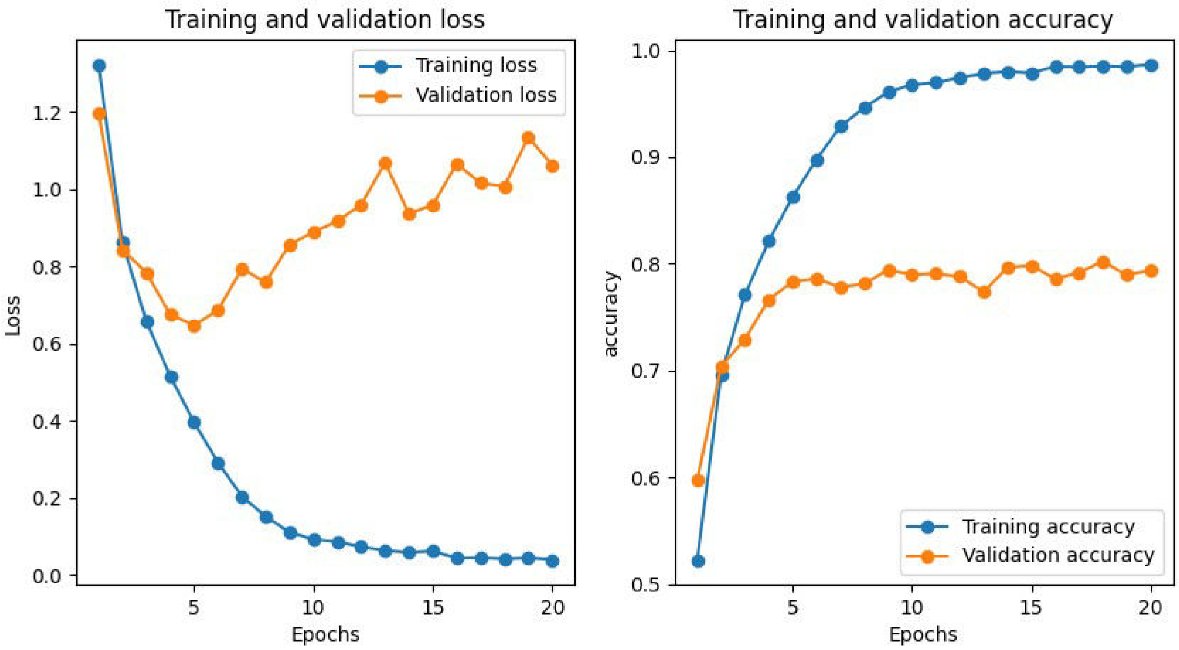
```
+    if data_augment:
+        dataset_transform = multiple_transform
+    else:
+        dataset_transform = base_transform
 
     if args.dataset == 'cifar10':
         # Download the train and test set and apply transform on it
-        train_set = datasets.CIFAR10(root='../data', train=True, download=True, transform=base_transform)
+        train_set = datasets.CIFAR10(root='../data', train=True, download=True,
transform=dataset_transform)
         test_set = datasets.CIFAR10(root='../data', train=False, download=True, transform=base_transform)
 
     elif args.dataset == 'svhn':
         # Download the train and test set and apply transform on it
-        train_set = datasets.SVHN(root='../data', split='train', download=True, transform=base_transform)
+        train_set = datasets.SVHN(root='../data', split='train', download=True,
transform=dataset_transform)
         test_set = datasets.SVHN(root='../data', split='test', download=True, transform=base_transform)
+
+    if val_set:
+        length_val_set = int(len(train_set)*val_set)
+        train_set, val_set = torch.utils.data.random_split(train_set, [len(train_set)-length_val_set,
length_val_set])
+        val_set.dataset = deepcopy(train_set.dataset)
+        val_set.dataset.transform = base_transform
 
     if args.optimizer == 'SGD':
         optimizer_factory = optimizer_setup(torch.optim.SGD, lr=learning_rate, momentum=0.9)
```

2. Courbes obtenues :

- CIFAR10 :

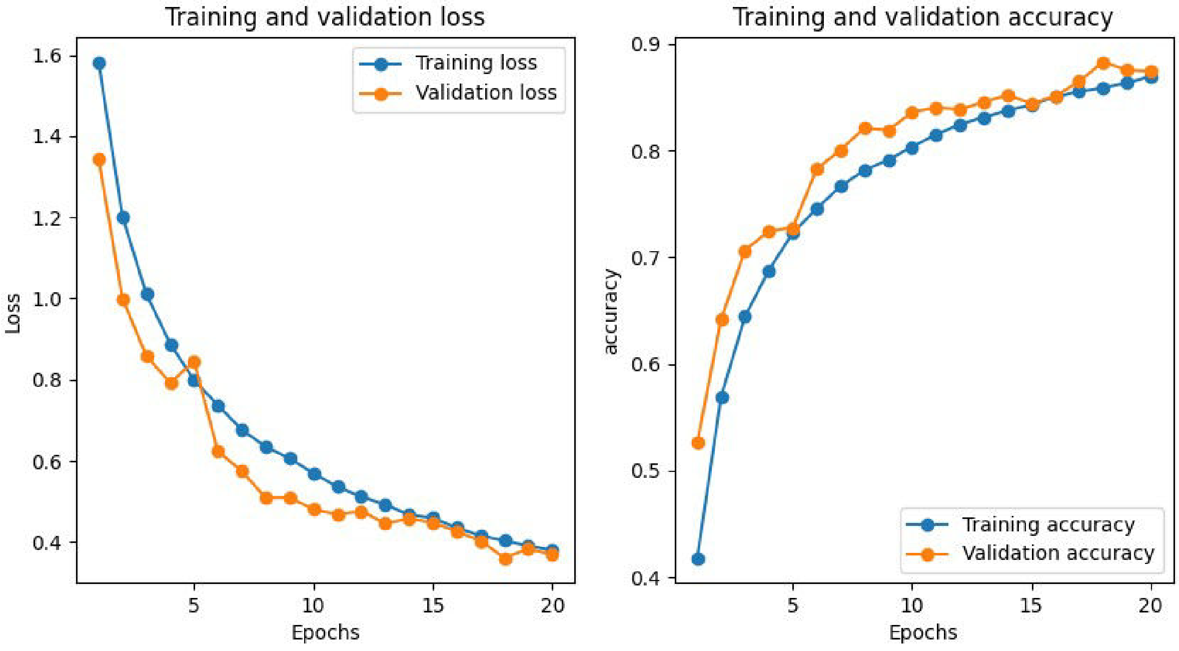
  ○ Sans augmentation des données :

```
!python train.py --model=ResNet --dataset=cifar10 --num-epochs=20 --batch_size=64
```



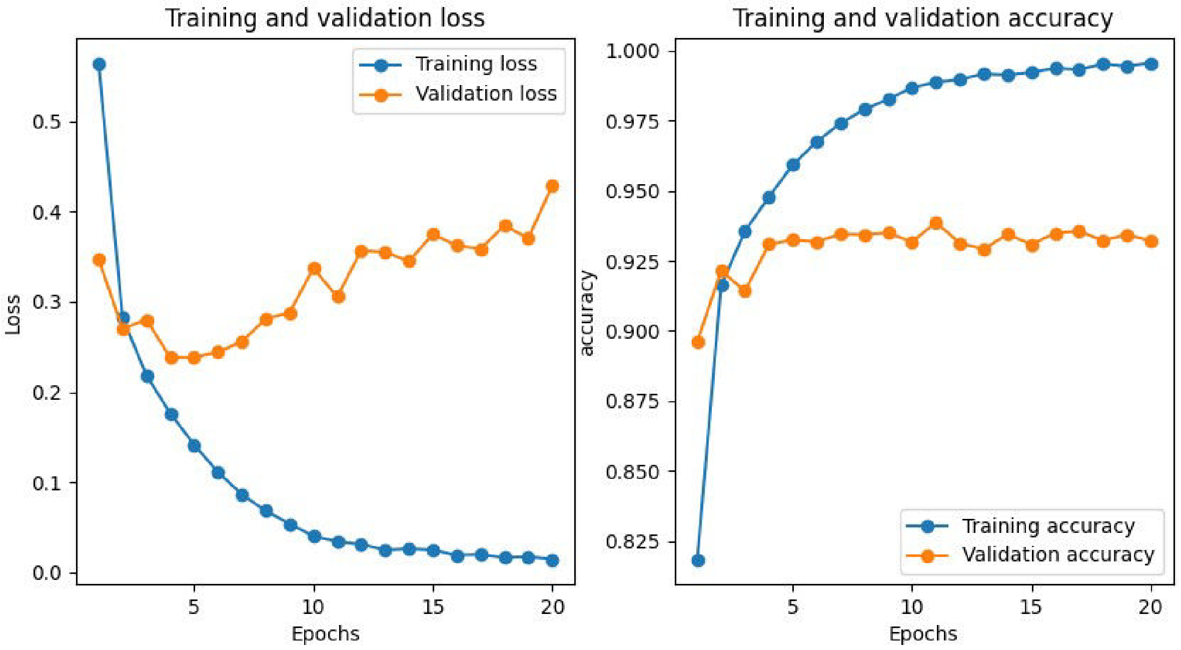
  ○ Avec augmentation des données :

```
!python train.py --model=ResNet --dataset=cifar10 --num-epochs=20 --data_aug --
batch_size=64
```



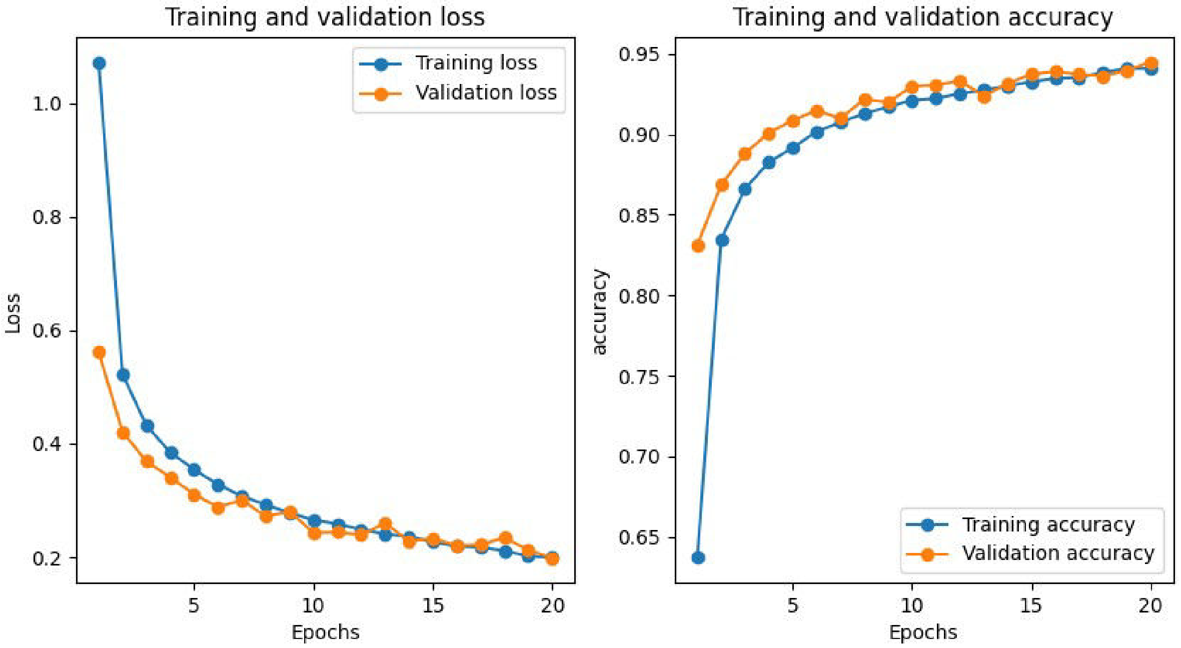
- SVHN :
  - Sans augmentation des données :

```
!python train.py --model=ResNet --dataset=svhn --num-epochs=20 -batch_size=64
```



  - Avec augmentation des données :

```
!python train.py --model=ResNet --dataset=svhn --num-epochs=20 --data_aug --
batch_size=64
```


Training and validation loss
Training loss
Validation loss
Loss
Epochs
Training and validation accuracy
accuracy
Training accuracy
Validation accuracy
Epochs