

IFT780 – TP3 – Question 2

Endroits où du code a été ajouté (texte en vert: ajout, texte en rouge: retrait)

- Fichier 'src/train.py':

```
[...]
+from copy import copy
+import numpy as np

[...]

    if data_augment:
        print('Data augmentation activated!')
+        data_augment_transforms = [
+            transforms.RandomRotation(15),
+            transforms.ColorJitter(contrast=0,
+                                   hue=0.1),
+            transforms.RandomHorizontalFlip(p=0.5),
+            transforms.RandomCrop(32, padding=4)
+        ]
    else:
        print('Data augmentation NOT activated!')
+        data_augment_transforms = []

[...]

-     base_transform = transforms.Compose([
-         transforms.ToTensor(),
-         transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))
-     ])
-
+     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)
-         test_set = datasets.CIFAR10(root='../data', train=False, download=True, transform=base_transform)
+         train_set = datasets.CIFAR10(root='../data', train=True, download=True, transform=transforms.ToTensor())
+         test_set = datasets.CIFAR10(root='../data', train=False, download=True, transform=None)
+
+     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)
-         test_set = datasets.SVHN(root='../data', split='test', download=True, transform=base_transform)
+         train_set = datasets.SVHN(root='../data', split='train', download=True, transform=transforms.ToTensor())
+         test_set = datasets.SVHN(root='../data', split='test', download=True, transform=None)
+
+     # Calculate dataset mean & std for normalization
+     print('Calculating dataset mean & standard deviation...')
+
+     r = []
+     g = []
+     b = []
+
+     for i in range(len(train_set)):
+         r.append(np.dstack(train_set[i][0][:, :, 0]))
+         g.append(np.dstack(train_set[i][0][:, :, 1]))
+         b.append(np.dstack(train_set[i][0][:, :, 2]))
+
+     mean = (np.mean(r), np.mean(g), np.mean(b))
+     std = (np.std(r), np.std(g), np.std(b))
+
+     train_transform = transforms.Compose([
+         transforms.ToTensor(),
+         transforms.Normalize(mean, std),
+         *data_augment_transforms
+     ])
+
+     base_transform = transforms.Compose([
+         transforms.ToTensor(),
+         transforms.Normalize(mean, std)
+     ])
+
+     train_set.transform = train_transform
+     test_set.transform = base_transform
+
+     if val_set:
+         len_val_set = int(len(train_set) * val_set)
+         train_set, val_set = torch.utils.data.random_split(train_set, [len(train_set) - len_val_set, len_val_set])
+         val_set.dataset = copy(train_set.dataset)
+         val_set.dataset.transform = base_transform

[...]
```

Courbes d'entraînement et de validation

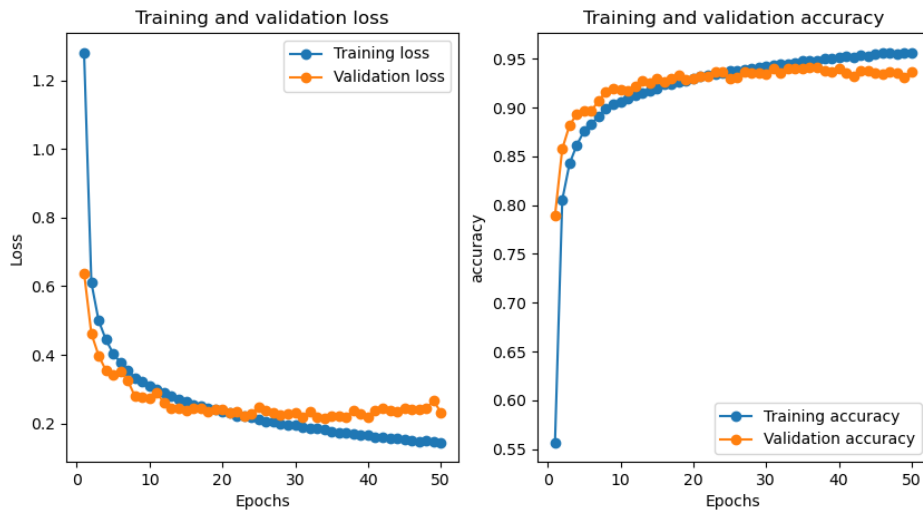
1. `--model=ResNet --dataset=svhn --num-epochs=50 --batch_size=100`



```
(ift725) simon@alien:~/tp3/src$ python train.py --model=ResNet --dataset=svhn --num-epochs=50 --batch_size=100
Data augmentation NOT activated!
Using downloaded and verified file: ../data/train_32x32.mat
Using downloaded and verified file: ../data/test_32x32.mat
Calculating dataset mean & standard deviation...
Training ResNet on svhn for 50 epochs
```

Finished training.
Accuracy (or Dice for UNet) on the test set: 93.282 %

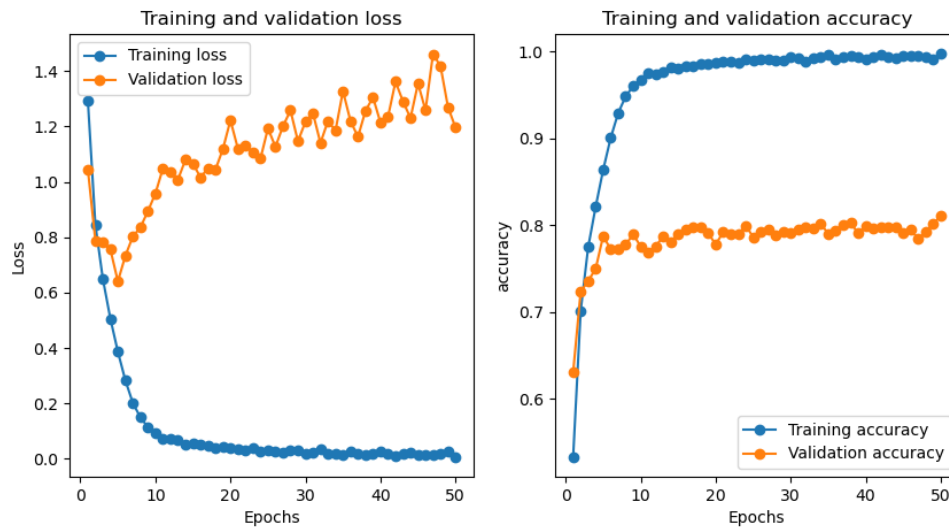
2. `--model=ResNet --dataset=svhn --num-epochs=50 --batch_size=100 --data_aug`



```
(ift725) simon@alien:~/tp3/src$ python train.py --model=ResNet --dataset=svhn --num-epochs=50 --batch_size=100 --data_aug
Data augmentation activated!
Using downloaded and verified file: ../data/train_32x32.mat
Using downloaded and verified file: ../data/test_32x32.mat
Calculating dataset mean & standard deviation...
Training ResNet on svhn for 50 epochs
```

Finished training.
Accuracy (or Dice for UNet) on the test set: 94.456 %

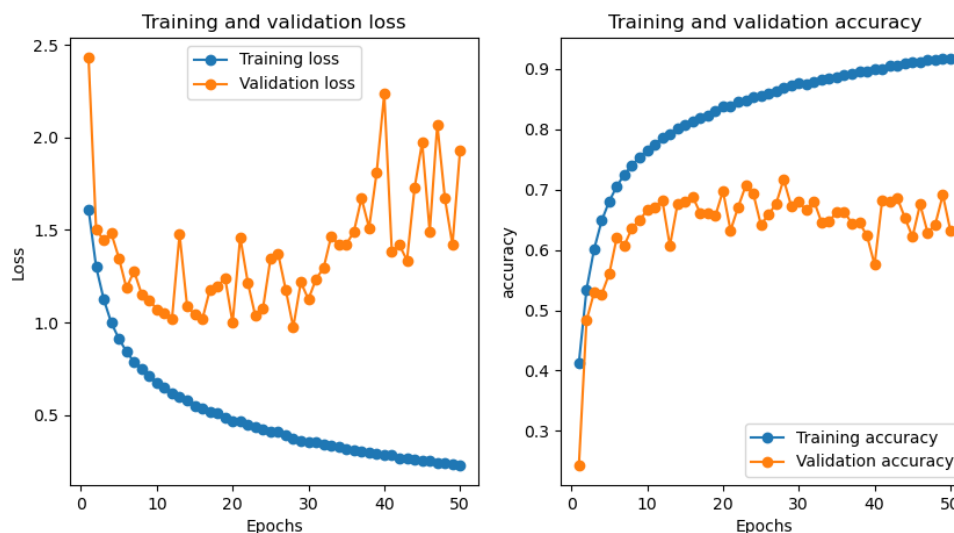
3. `--model=ResNet --dataset=cifar10 --num-epochs=50 --batch_size=100`



```
(ift725) simon@alien:~/tp3/src$ python train.py --model=ResNet --dataset=cifar10 --num-epochs=50 --batch_size=100
Data augmentation NOT activated!
Files already downloaded and verified
Files already downloaded and verified
Calculating dataset mean & standard deviation...
Training ResNet on cifar10 for 50 epochs
```

Finished training.
Accuracy (or Dice for UNet) on the test set: 80.180 %

4. `--model=ResNet --dataset=cifar10 --num-epochs=50 --batch_size=100 --data_aug`



```
(ift725) simon@alien:~/tp3/src$ python train.py --model=ResNet --dataset=cifar10 --num-epochs=50 --batch_size=100 --data_aug
Data augmentation activated!
Files already downloaded and verified
Files already downloaded and verified
Calculating dataset mean & standard deviation...
Training ResNet on cifar10 for 50 epochs
```

Finished training.
Accuracy (or Dice for UNet) on the test set: 85.580 %