MNIST Diffusion Project

Generated by Doxygen 1.9.1

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 ddpm_train Namespace Reference

Classes

- class CNNBlock
- class CNN
- class DDPM

Functions

- Dict[str, torch.Tensor] ddpm_schedules (float beta1, float beta2, int T)

  Returns pre-computed schedules for DDPM sampling with a linear noise schedule.

Variables

- tf = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5,), (1.0))])
- dataset = MNIST("./data", train=True, download=True, transform=tf)
- dataloader = DataLoader(dataset, batch_size=128, shuffle=True, num_workers=4, drop_last=True)
- gt = CNN(in_channels=1, expected_shape=(28, 28), n_hidden=(16, 32, 32, 16), act=nn.GELU)
- ddpm = DDPM(gt=gt, betas=(1e-4, 0.02), n_T=1000)
- optim = torch.optim.Adam(ddpm.parameters(), lr=2e-4)
- accelerator = Accelerator()
- int n_epoch = 100
- list losses = [ ]
- pbar = tqdm(dataloader)
- loss = ddpm(x)
- avg_loss = np.average(losses[min(len(losses)-100, 0):])
- xh = ddpm.sample(16, (1, 28, 28), accelerator.device)
- grid = make_grid(xh, nrow=4)

5.1.1 Function Documentation

5.1.1.1 ddpm_schedules()

```
Dict[str, torch.Tensor] ddpm_train.ddpm_schedules (
             float beta1,
             float beta2,
             int T )
```

Returns pre-computed schedules for DDPM sampling with a linear noise schedule.

5.1.2 Variable Documentation

5.1.2.1 accelerator

```
ddpm_train.accelerator = Accelerator()
```

5.1.2.2 avg_loss

```
ddpm_train.avg_loss = np.average(losses[min(len(losses)-100, 0):])
```

5.1.2.3 dataloader

```
ddpm_train.dataloader = DataLoader(dataset, batch_size=128, shuffle=True, num_workers=4, drop↩
_last=True)
```

5.1.2.4 dataset

```
ddpm_train.dataset = MNIST("./data", train=True, download=True, transform=tf)
```

5.1.2.5 ddpm

```
ddpm_train.ddpm = DDPM(gt=gt, betas=(1e-4, 0.02), n_T=1000)
```

5.1.2.6 grid

```
ddpm_train.grid = make_grid(xh, nrow=4)
```

5.1.2.7 gt

```
ddpm_train.gt = CNN(in_channels=1, expected_shape=(28, 28), n_hidden=(16, 32, 32, 16), act=nn.↩
GELU)
```

5.1.2.8 loss

```
ddpm_train.loss = ddpm(x)
```

5.1.2.9 losses

```
list ddpm_train.losses = [ ]
```

5.1.2.10 n_epoch

```
int ddpm_train.n_epoch = 100
```

5.1.2.11 optim

```
ddpm_train.optim = torch.optim.Adam(ddpm.parameters(), lr=2e-4)
```

5.1.2.12 pbar

```
ddpm_train.pbar = tqdm(dataloader)
```

5.1.2.13 tf

```
ddpm_train.tf = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5,), (1.↩
0))])
```

5.1.2.14 xh

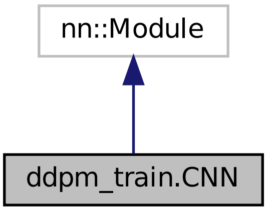
```
ddpm_train.xh = ddpm.sample(16, (1, 28, 28), accelerator.device)
```

Chapter 6

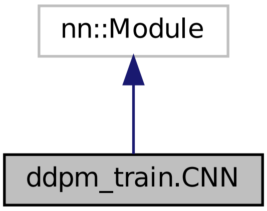
Class Documentation

6.1 ddpm_train.CNN Class Reference

Inheritance diagram for ddpm_train.CNN:



Collaboration diagram for ddpm_train.CNN:



Public Member Functions

- None __init__ (self, in_channels, expected_shape=(28, 28), n_hidden=(64, 128, 64), kernel_size=7, last_kernel_size=3, time_embeddings=16, act=nn.GELU)
- torch.Tensor time_encoding (self, torch.Tensor t)
- torch.Tensor forward (self, torch.Tensor x, torch.Tensor t)

Public Attributes

- blocks
- time_embed

6.1.1 Constructor & Destructor Documentation

6.1.1.1 __init__()

```
None ddpm_train.CNN.__init__ (
              self,
              in_channels,
              expected_shape = (28, 28),
              n_hidden = (64, 128, 64),
              kernel_size = 7,
              last_kernel_size = 3,
              time_embeddings = 16,
              act = nn.GELU )
```

6.1.2 Member Function Documentation

6.1.2.1 forward()

```
torch.Tensor ddpm_train.CNN.forward (
              self,
             torch.Tensor x,
             torch.Tensor t )
```

6.1.2.2 time_encoding()

```
torch.Tensor ddpm_train.CNN.time_encoding (
              self,
             torch.Tensor t )
```

6.1.3 Member Data Documentation

6.1.3.1 blocks

```
ddpm_train.CNN.blocks
```

6.1.3.2 time_embed

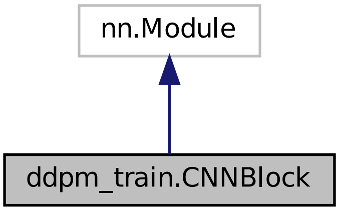
```
ddpm_train.CNN.time_embed
```

The documentation for this class was generated from the following file:

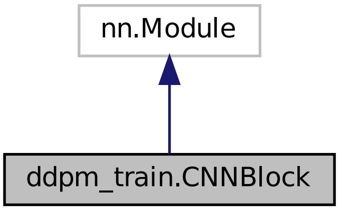
- /home/jhughes2712/projects/m2_assessment/jh2284/src/ddpm_train.py

6.2 ddpm_train.CNNBlock Class Reference

Inheritance diagram for ddpm_train.CNNBlock:



Collaboration diagram for ddpm_train.CNNBlock:



Public Member Functions

- def __init__ (self, in_channels, out_channels, ∗expected_shape, act=nn.GELU, kernel_size=7)
- def forward (self, x)

Public Attributes

- net

6.2.1 Constructor & Destructor Documentation

6.2.1.1 __init__()

```
def ddpm_train.CNNBlock.__init__ (
              self,
              in_channels,
              out_channels,
             ∗ expected_shape,
              act = nn.GELU,
              kernel_size = 7 )
```

6.2.2 Member Function Documentation

6.2.2.1 forward()

```
def ddpm_train.CNNBlock.forward (
              self,
              x )
```

6.2.3 Member Data Documentation

6.2.3.1 net

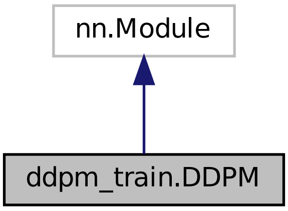
```
ddpm_train.CNNBlock.net
```

The documentation for this class was generated from the following file:

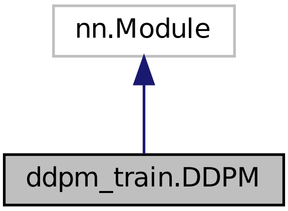
- /home/jhughes2712/projects/m2_assessment/jh2284/src/ddpm_train.py

6.3 ddpm_train.DDPM Class Reference

Inheritance diagram for ddpm_train.DDPM:



Collaboration diagram for ddpm_train.DDPM:



Public Member Functions

- None __init__ (self, gt, Tuple[float, float] betas, int n_T, nn.Module criterion=nn.MSELoss())
- torch.Tensor forward (self, torch.Tensor x)

  Algorithm 18.1 in Prince.
- torch.Tensor sample (self, int n_sample, size, device)

  Algorithm 18.2 in Prince.

Public Attributes

- gt
- n_T
- criterion

6.3.1 Constructor & Destructor Documentation

6.3.1.1 __init__()

```
None ddpm_train.DDPM.__init__ (
              self,
              gt,
             Tuple[float, float] betas,
             int n_T,
             nn.Module  criterion = nn.MSELoss() )
```

6.3.2 Member Function Documentation

6.3.2.1 forward()

```
torch.Tensor ddpm_train.DDPM.forward (
              self,
             torch.Tensor x )
```

Algorithm 18.1 in Prince.

6.3.2.2 sample()

```
torch.Tensor ddpm_train.DDPM.sample (
              self,
             int n_sample,
              size,
              device )
```

Algorithm 18.2 in Prince.

6.3.3 Member Data Documentation

6.3.3.1 criterion

```
ddpm_train.DDPM.criterion
```

6.3.3.2 gt

```
ddpm_train.DDPM.gt
```

6.3.3.3 n_T

```
ddpm_train.DDPM.n_T
```

The documentation for this class was generated from the following file:

- /home/jhughes2712/projects/m2_assessment/jh2284/src/ddpm_train.py

Chapter 7

File Documentation

7.1 /home/jhughes2712/projects/m2_assessment/jh2284/src/ddpm_↩ train.py File Reference

Runs DDPM diffusion model.

Classes

- class ddpm_train.CNNBlock
- class ddpm_train.CNN
- class ddpm_train.DDPM

Namespaces

- ddpm_train

Functions

- Dict[str, torch.Tensor] ddpm_train.ddpm_schedules (float beta1, float beta2, int T)

  Returns pre-computed schedules for DDPM sampling with a linear noise schedule.

Variables

- ddpm_train.tf = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.5,), (1.0))])
- ddpm_train.dataset = MNIST("./data", train=True, download=True, transform=tf)
- ddpm_train.dataloader = DataLoader(dataset, batch_size=128, shuffle=True, num_workers=4, drop_↩ last=True)
- ddpm_train.gt = CNN(in_channels=1, expected_shape=(28, 28), n_hidden=(16, 32, 32, 16), act=nn.GELU)
- ddpm_train.ddpm = DDPM(gt=gt, betas=(1e-4, 0.02), n_T=1000)
- ddpm_train.optim = torch.optim.Adam(ddpm.parameters(), lr=2e-4)
- ddpm_train.accelerator = Accelerator()
- int ddpm_train.n_epoch = 100
- list ddpm_train.losses = [ ]
- ddpm_train.pbar = tqdm(dataloader)
- ddpm_train.loss = ddpm(x)
- ddpm_train.avg_loss = np.average(losses[min(len(losses)-100, 0):])
- ddpm_train.xh = ddpm.sample(16, (1, 28, 28), accelerator.device)
- ddpm_train.grid = make_grid(xh, nrow=4)

7.1.1 Detailed Description

Runs DDPM diffusion model.

Author

Created by J. Hughes on 18/03/2024.

Index