vae

January 27, 2020

[1]:

```
import itertools
import random
import matplotlib.pyplot as plt
import torchvision
import torch
import torch.utils.data
import torch.nn as nn
import numpy as np

from typing import Any
from scipy.stats import norm
from torch.utils.tensorboard import SummaryWriter
from torchvision import transforms

from src.py.models.utils.utils import learn
from src.py.models.vae import VAE
from src.py.utils.digit_plotting import plot_digits, draw_random_picture
```

[2]:

```
DATA_PATH = "../../data"
BATCH_SIZE = 10
DIGIT_SIZE = 28
```

[3]:

```
trans = transforms.Compose([transforms.ToTensor()]) #, transforms.Normalize((0.
 ↪5,), (1.0,))])
train_data = torchvision.datasets.MNIST(DATA_PATH, transform=trans, train=True,␣
 ↪download=True)
test_data = torchvision.datasets.MNIST(DATA_PATH, transform=trans, train=False,␣
 ↪download=True)

train_loader = torch.utils.data.DataLoader(dataset=train_data,␣
 ↪batch_size=BATCH_SIZE)
```

[4]:

```
with SummaryWriter(log_dir="../../runs/vae", purge_step=0) as writer:
    vae = VAE(writer)
    vae = learn(vae, train_loader, 3)
```

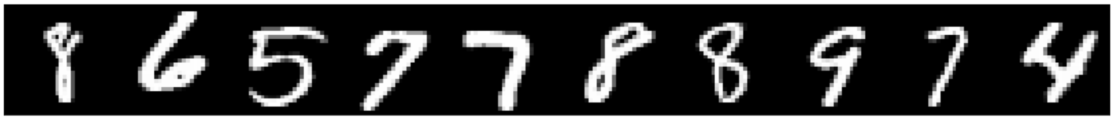
[5]:

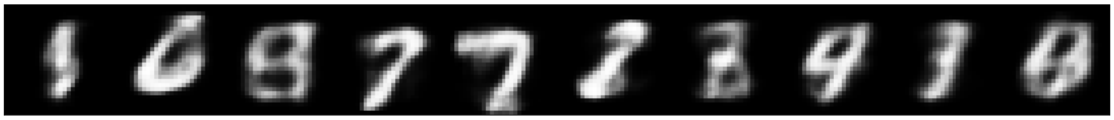
```
def plot_result_random_example_batch_from_head(head_size=100):
    X, y = random.sample(list(itertools.islice(iter(train_loader),
                                               None, head_size)), 1)[0]

    r = vae.auto_encoder(X)
    plot_digits(X)
    plot_digits(r)
```

[6]:

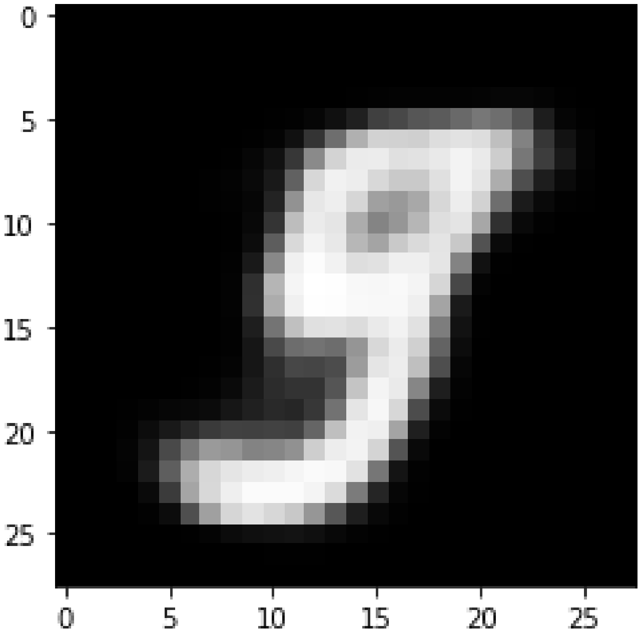
```
plot_result_random_example_batch_from_head()
```





[7]:

```
_, decoder = vae.get_models()
draw_random_picture(decoder)
```


0
5
10
15
20
25
0
5
10
15
20
25