Conv - AutoEncoder implementation

created by [Etzion Harari (https://github.com/EtzionR)](https://github.com/EtzionR)

Full documantation: [https://github.com/EtzionR/My-TF-AutoEncoder (https://github.com/EtzionR/My-TF-AutoEncoder)](https://github.com/EtzionR/My-TF-AutoEncoder)

Load helpful AutoEncoder & libraries:

In [21]:

```
from cec import AutoEncoder

import matplotlib.pyplot as plt
import numpy as np
```

Load & Training model:

In [22]:

```
# load data
trn = np.load(r'data\mnist_trn.npy')

# define variables
source = (28,28)
kernels = [5,5]
filters = 15
latant_dim = 2
epochs = 50
lr = 0.001

# using the code
aec = AutoEncoder(source=source,kernels=kernels,filters=filters,latant_dim=latant_dim,epochs=epochs,lr=lr)

# fitting the model
aec.fit(trn,trn)
```

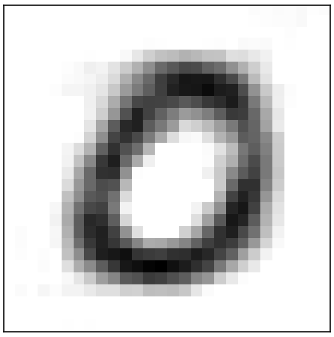
```
Epoch 41/50
32/32 [==============================] - 1s 28ms/step - loss: 3109.6887
Epoch 42/50
32/32 [==============================] - 1s 28ms/step - loss: 3087.2371
Epoch 43/50
32/32 [==============================] - 1s 28ms/step - loss: 3093.1353
Epoch 44/50
32/32 [==============================] - 1s 28ms/step - loss: 3088.5125
Epoch 45/50
32/32 [==============================] - 1s 28ms/step - loss: 3081.7444
Epoch 46/50
32/32 [==============================] - 1s 28ms/step - loss: 3067.9253
Epoch 47/50
32/32 [==============================] - 1s 28ms/step - loss: 3069.9802
Epoch 48/50
32/32 [==============================] - 1s 28ms/step - loss: 3051.2183
Epoch 49/50
32/32 [==============================] - 1s 28ms/step - loss: 3067.1235 0s - los
Epoch 50/50
32/32 [==============================] - 1s 28ms/step - loss: 3062.9741
```

plot one prediction output example:

In [23]:

```
# get prediction
plt.figure(figsize=(4,4))
plt.imshow(aec.predict(trn)[1],cmap='binary')
plt.xticks([])
plt.yticks([])
plt.show()
```

```
WARNING:tensorflow:Model was constructed with shape (None, 1, 2) for input Tensor("input_7:0", shape=(None, 1, 2), dtype
=float32), but it was called on an input with incompatible shape (None, 2).
```

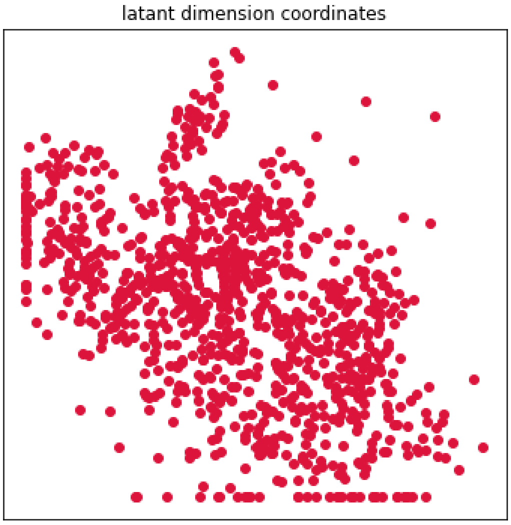


plot latant dimension encoded coordinates

In [28]:

```
coords = np.array(aec.encode(trn))

plt.figure(figsize=(6,6))
plt.title('latant dimension coordinates')
plt.scatter(coords[:,0],coords[:,1],color='crimson')
plt.xticks([])
plt.yticks([])
plt.show()
```



latant dimension coordinates

noise filtering

In [25]:

```
unoisy  = np.load(r'data\unoisy.npy')
noisy   = np.load(r'data\noisy.npy')

noise_f = AutoEncoder(source=source,kernels=kernels,filters=3,latant_dim=100,epochs=50,lr=lr)

noise_f.fit(noisy[:200],unoisy[:200])
```

```
Epoch 42/50
7/7 [==============================] - 0s 8ms/step - loss: 1357.8938
Epoch 43/50
7/7 [==============================] - 0s 8ms/step - loss: 1346.9268
Epoch 44/50
7/7 [==============================] - 0s 8ms/step - loss: 1329.8282
Epoch 45/50
7/7 [==============================] - 0s 8ms/step - loss: 1319.1084
Epoch 46/50
7/7 [==============================] - 0s 8ms/step - loss: 1308.0422
Epoch 47/50
7/7 [==============================] - 0s 8ms/step - loss: 1288.2487
Epoch 48/50
7/7 [==============================] - 0s 8ms/step - loss: 1276.2523
Epoch 49/50
7/7 [==============================] - 0s 8ms/step - loss: 1260.5668
Epoch 50/50
7/7 [==============================] - 0s 8ms/step - loss: 1246.8910
```

Out[25]:

```
<cec.AutoEncoder at 0x253062efac0>
```

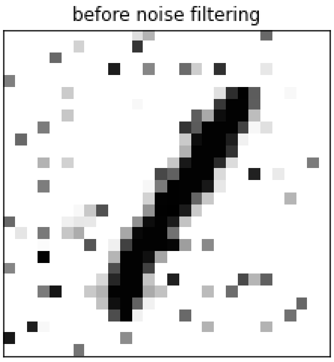
plot the digit after noise filtering

In [29]:

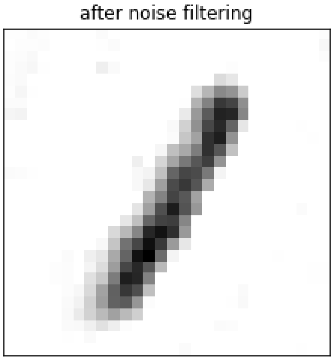
```
plt.figure(figsize=(4,4))
plt.title('before noise filtering')
plt.imshow(noisy[201],cmap='binary')
plt.xticks([])
plt.yticks([])
plt.show()

img = noise_f.predict(noisy[201:])[0]

plt.figure(figsize=(4,4))
plt.title('after noise filtering')
plt.imshow(img,cmap='binary')
plt.xticks([])
plt.yticks([])
plt.show()
```



before noise filtering



after noise filtering