

```
In [1]: import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
import sys
sys.path.append("../py3_venv/lib/python3.5/site-packages/")
from MulticoreTSNE import MulticoreTSNE as TSNE
import umap.umap_ as umap
from keras.preprocessing.sequence import pad_sequences
from keras.datasets import mnist, fashion_mnist, imdb
import matplotlib
from matplotlib import pyplot as plt
import time
import os
os.environ["CUDA_VISIBLE_DEVICES"]=""
```

```
/usr/local/lib/python3.5/dist-packages/h5py/__init__.py:34: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

```
In [2]: plt.rcParams['figure.figsize'] = 15, 15
```

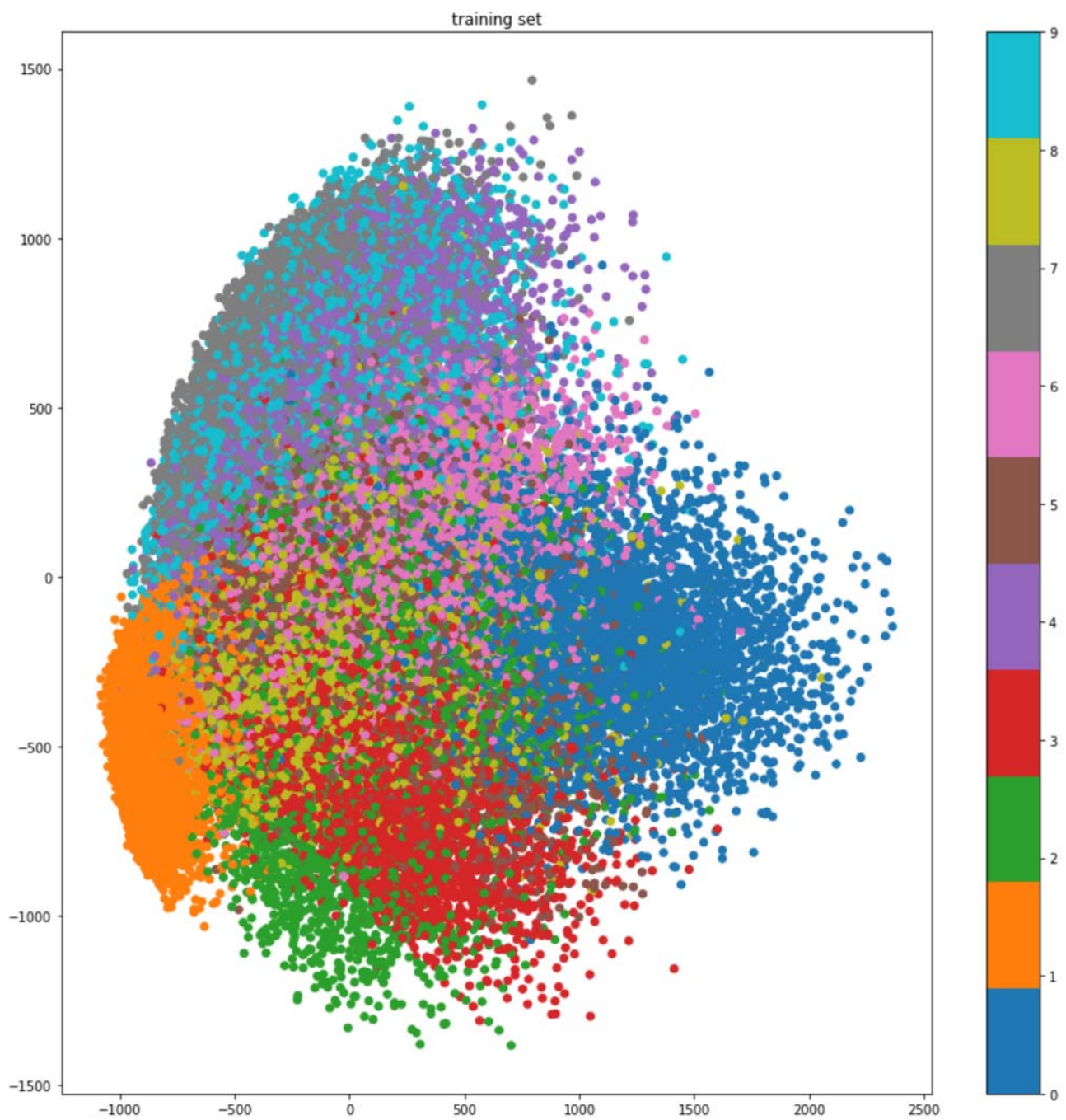
```
In [3]: (x_train, y_train), (x_test, y_test) = mnist.load_data()
```

```
In [4]: start = time.time()
pca_mod = PCA()
pca_mod.fit(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

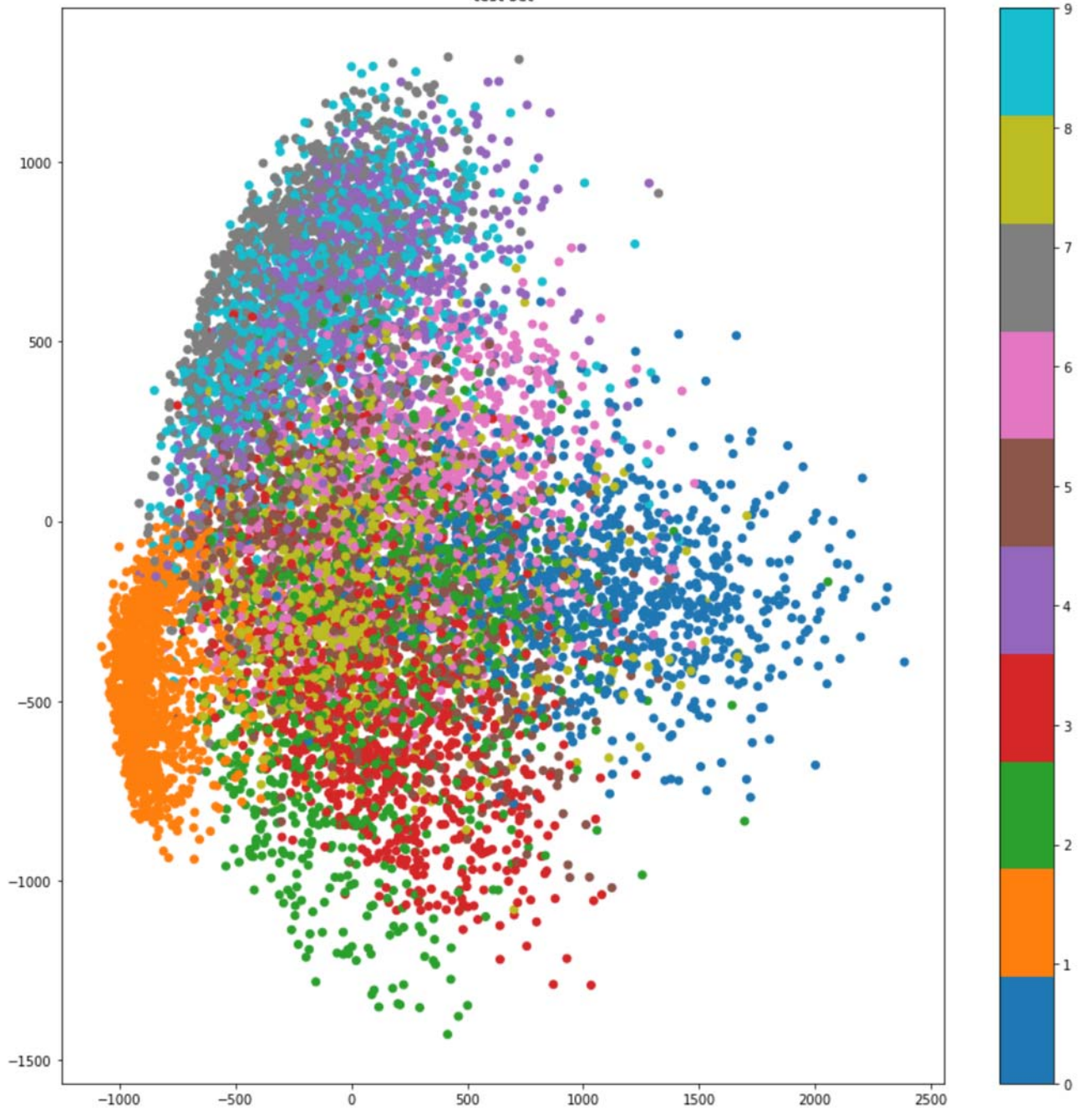
pca_res = pca_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(pca_res[:,0], pca_res[:,1], cmap=plt.cm.tab10, c=y_train)
cbar= plt.colorbar()
plt.title('training set')
plt.show()

pca_res = pca_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(pca_res[:,0], pca_res[:,1], cmap=plt.cm.tab10, c=y_test)
cbar= plt.colorbar()
plt.title('test set')
plt.show()
```

00:00:16



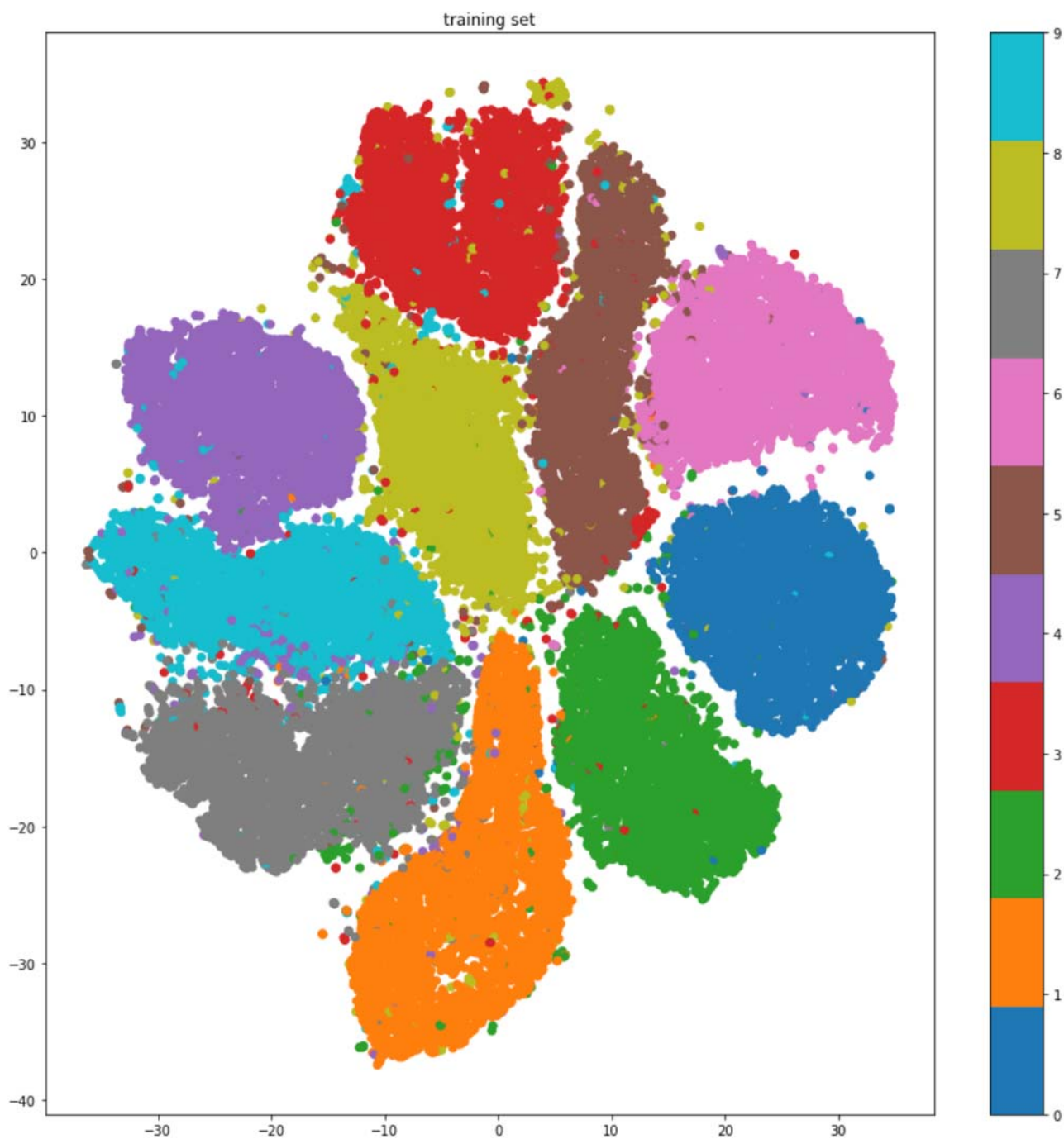
test set



```
In [5]: start = time.time()
tsne_res = TSNE(n_jobs=16).fit_transform(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

plt.scatter(tsne_res[:,0], tsne_res[:,1], cmap=plt.cm.tab10, c=y_train)
cbar= plt.colorbar()
plt.title('training set')
plt.show()
```

00:09:46

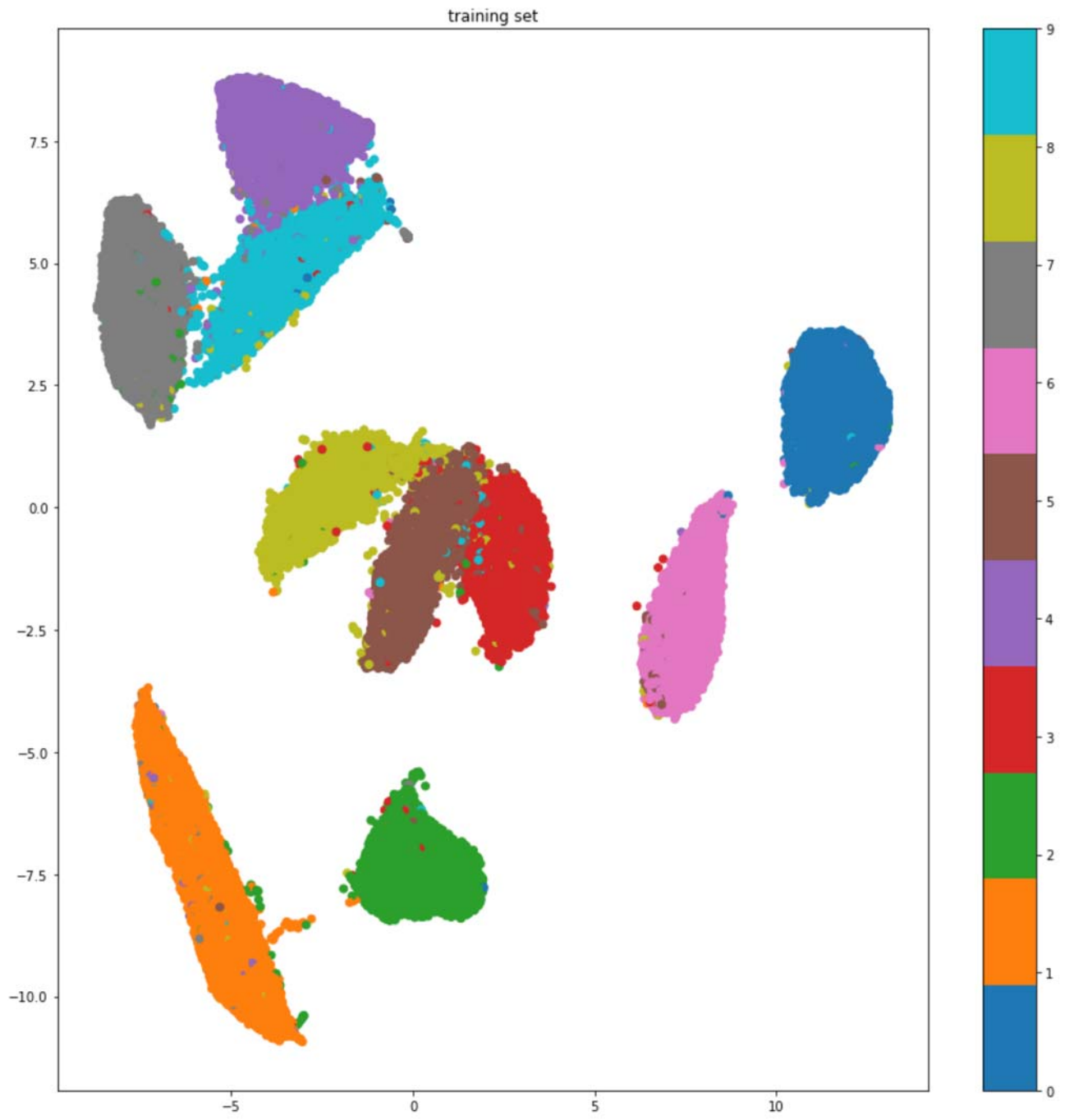


```
In [6]: start = time.time()
umap_mod = umap.UMAP()
umap_mod.fit(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

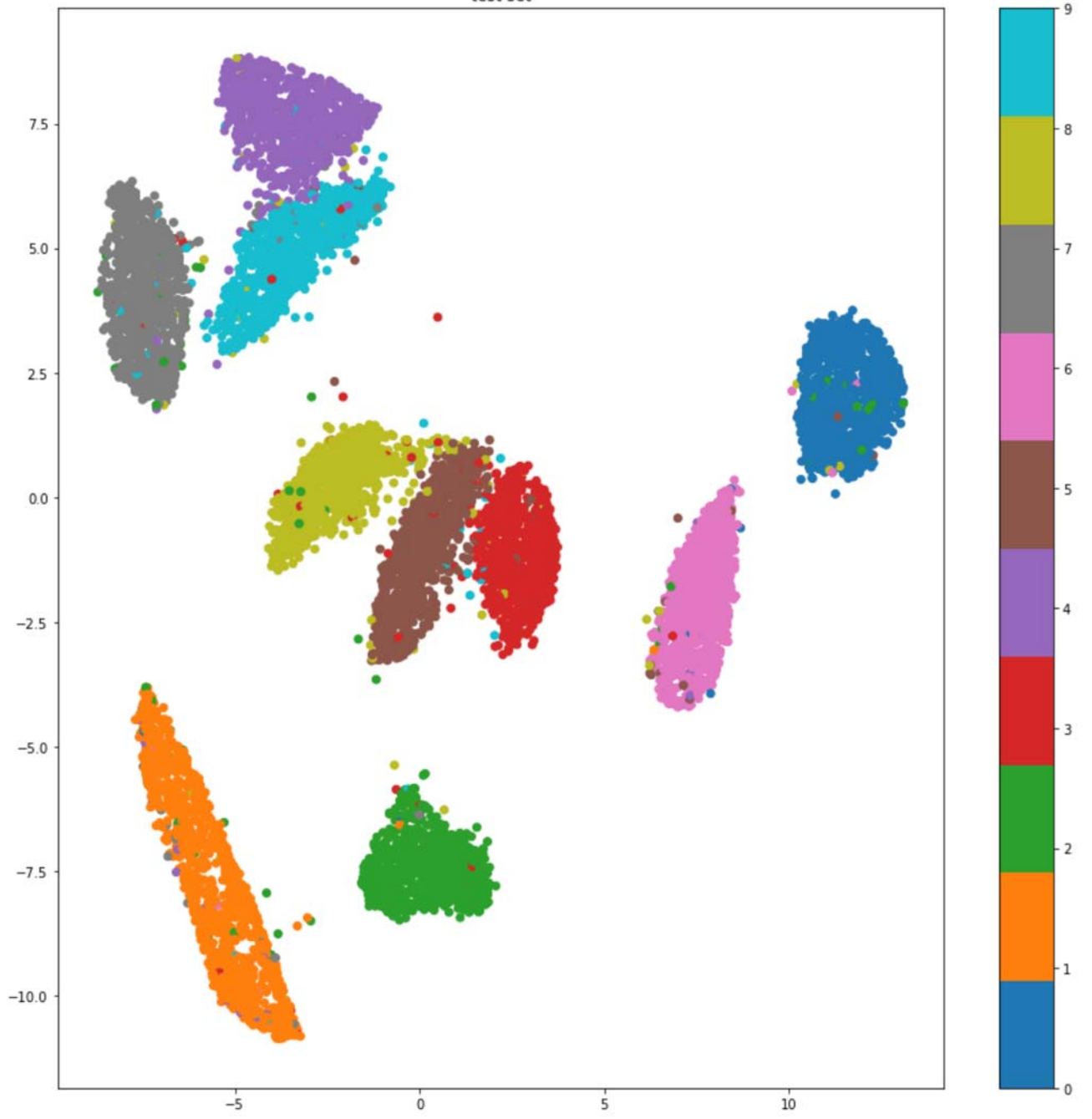
umap_res = umap_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_train)
cbar= plt.colorbar()
plt.title('training set')
plt.show()

umap_res = umap_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_test)
cbar= plt.colorbar()
plt.title('test set')
plt.show()
```





test set

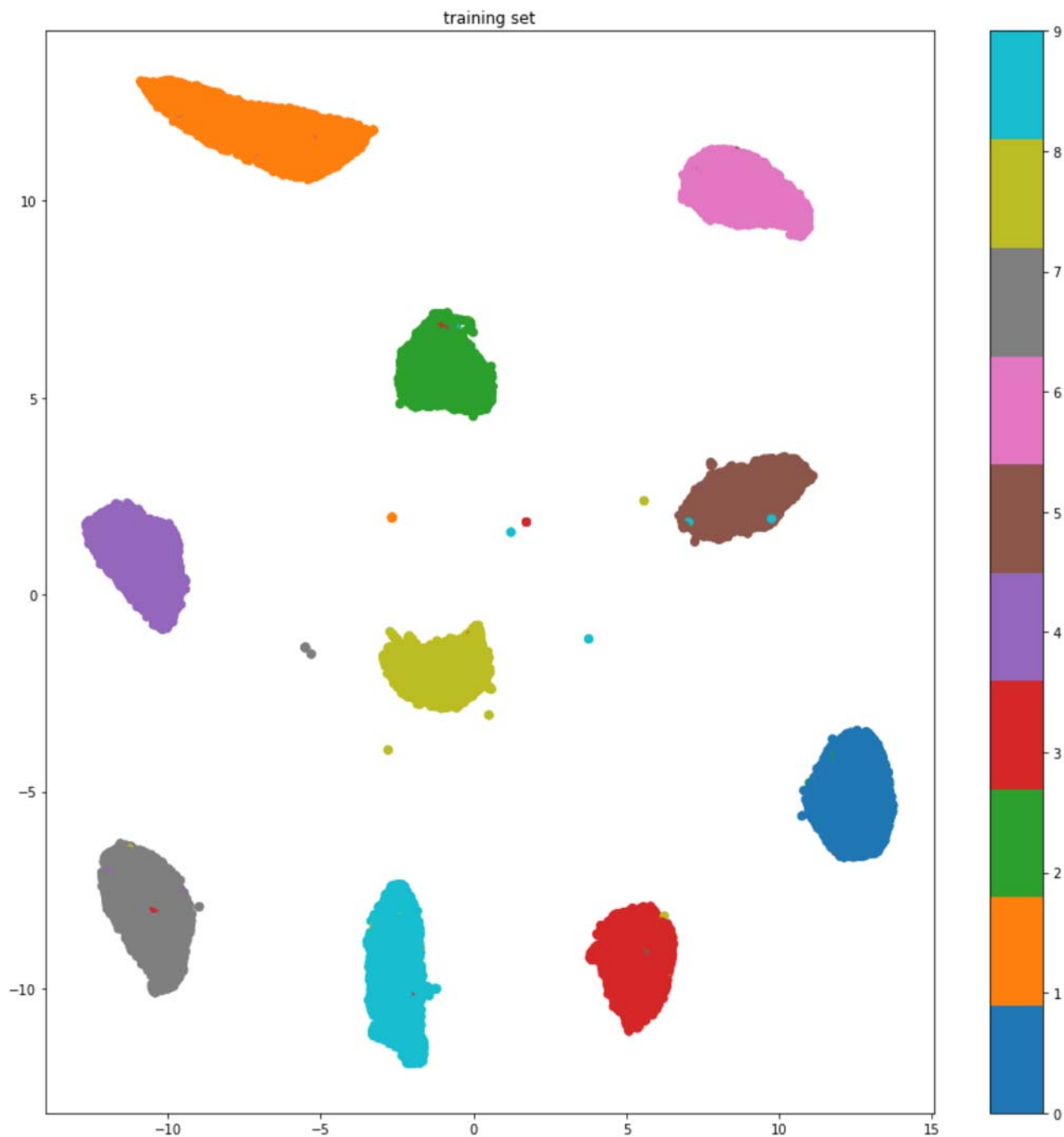


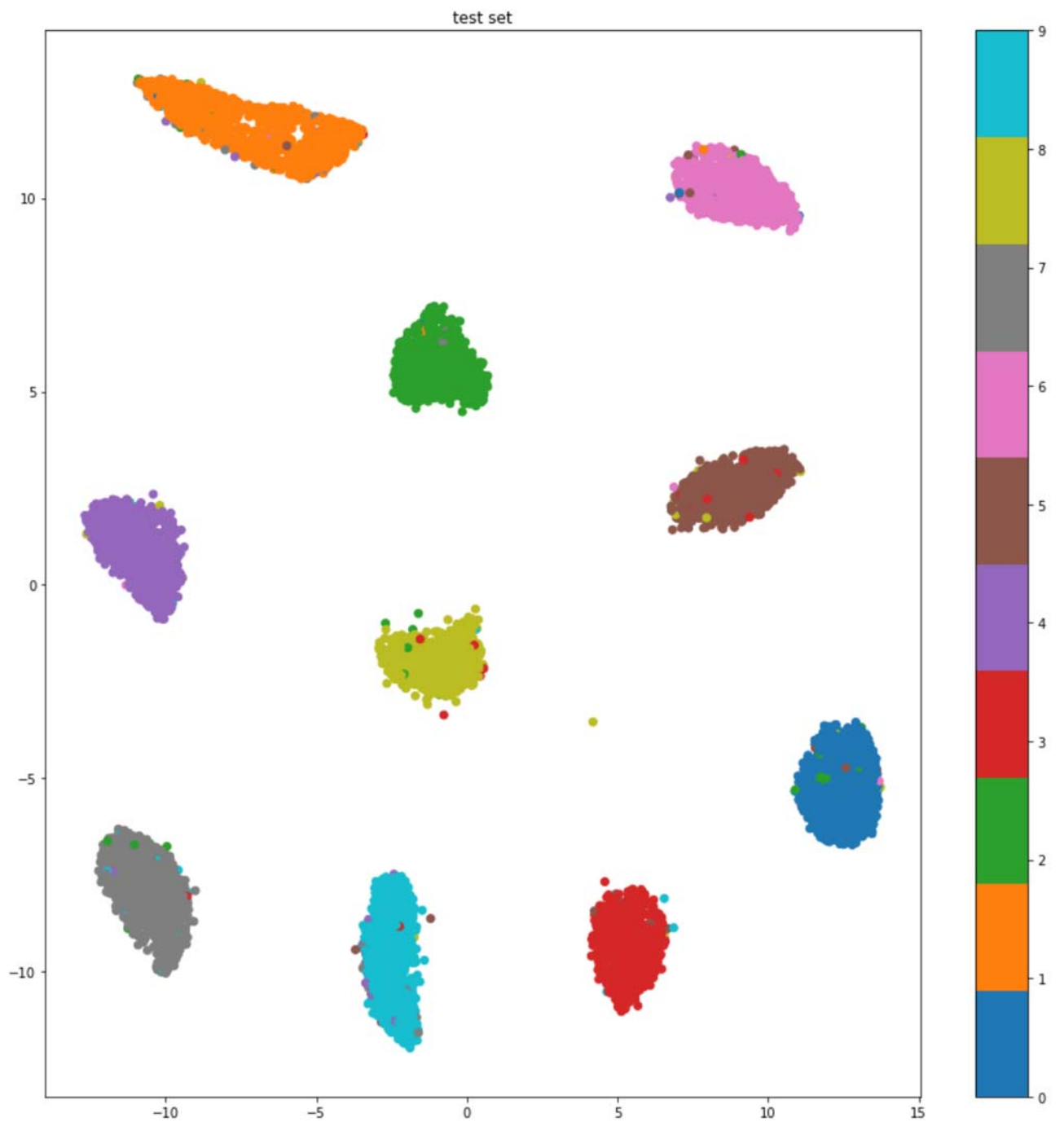


```
In [7]: start = time.time()
umap_mod = umap.UMAP()
umap_mod.fit(x_train.reshape((x_train.shape[0],-1)),y_train)
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

umap_res = umap_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_train)
cbar= plt.colorbar()
plt.title('training set')
plt.show()

umap_res = umap_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_test)
cbar= plt.colorbar()
plt.title('test set')
plt.show()
```





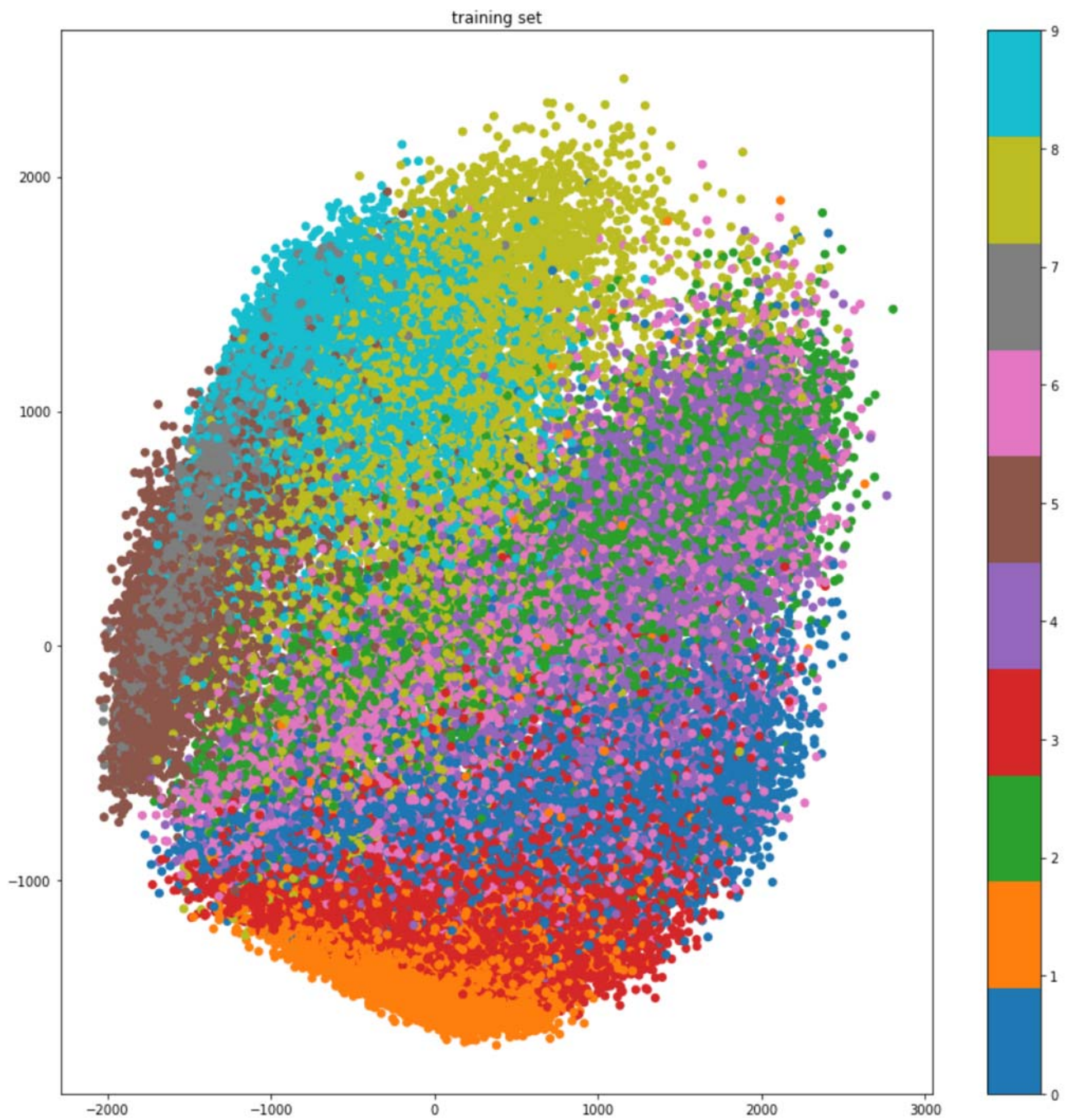
```
In [8]: (x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

```
In [ ]: start = time.time()
pca_mod = PCA()
pca_mod.fit(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

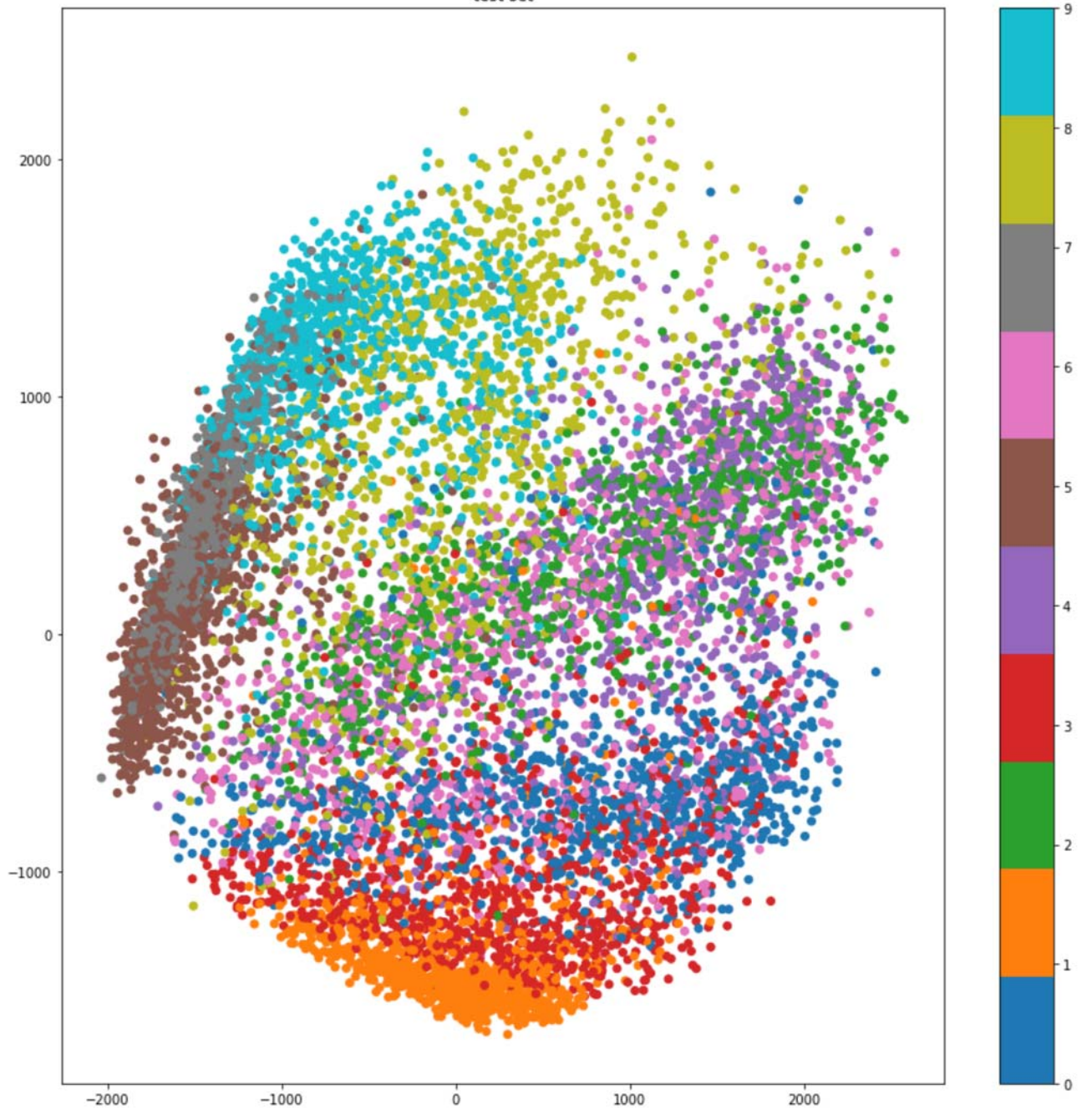
pca_res = pca_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(pca_res[:,0], pca_res[:,1], cmap=plt.cm.tab10, c=y_train)
cbar= plt.colorbar()
plt.title('training set')
plt.show()

pca_res = pca_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(pca_res[:,0], pca_res[:,1], cmap=plt.cm.tab10, c=y_test)
cbar= plt.colorbar()
plt.title('test set')
plt.show()
```

00:00:16



test set

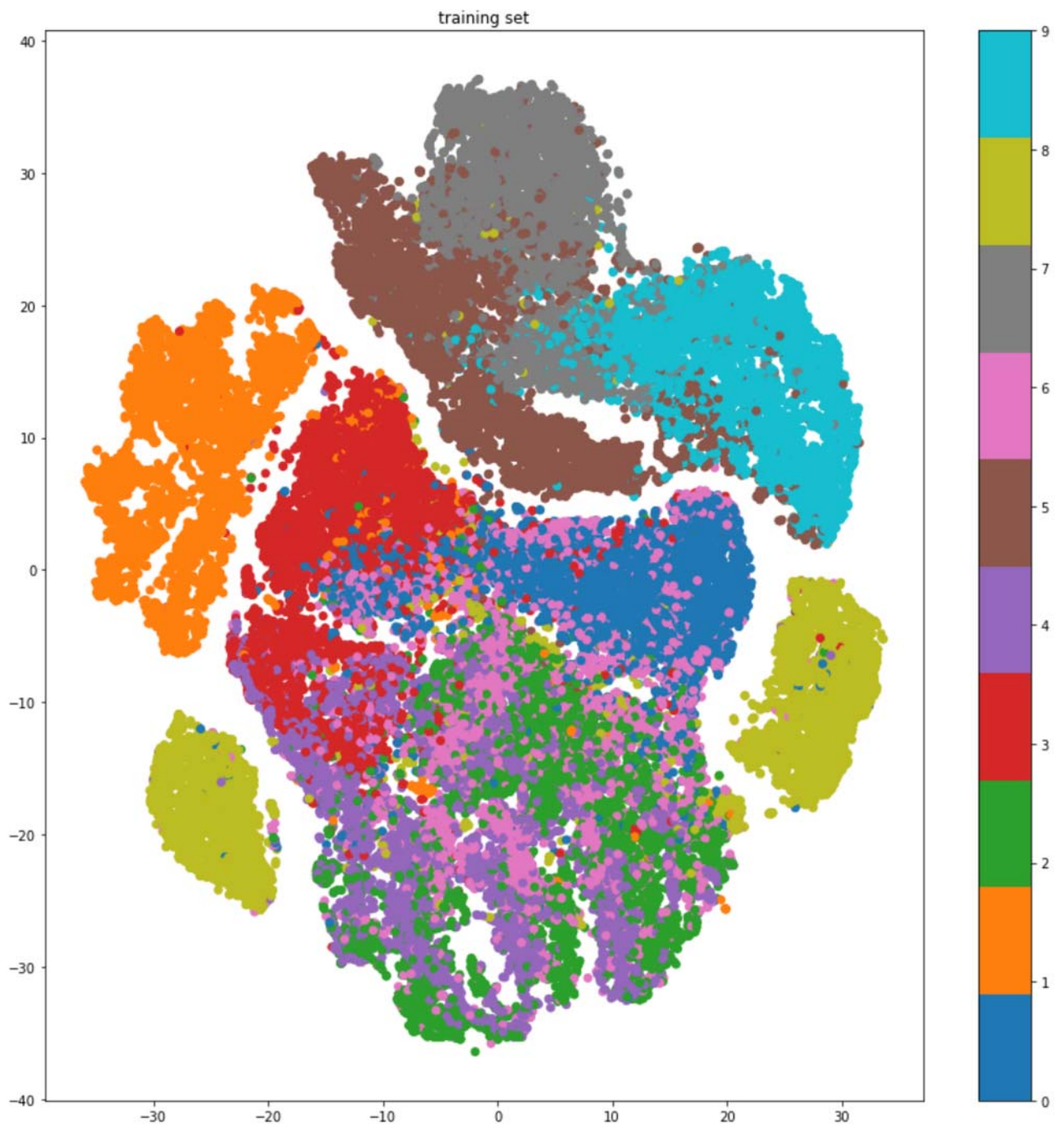




```
In [ ]: start = time.time()
tsne_res = TSNE(n_jobs=16).fit_transform(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

plt.scatter(tsne_res[:,0], tsne_res[:,1], cmap=plt.cm.tab10, c=y_train)
cbar= plt.colorbar()
plt.title('training set')
plt.show()
```

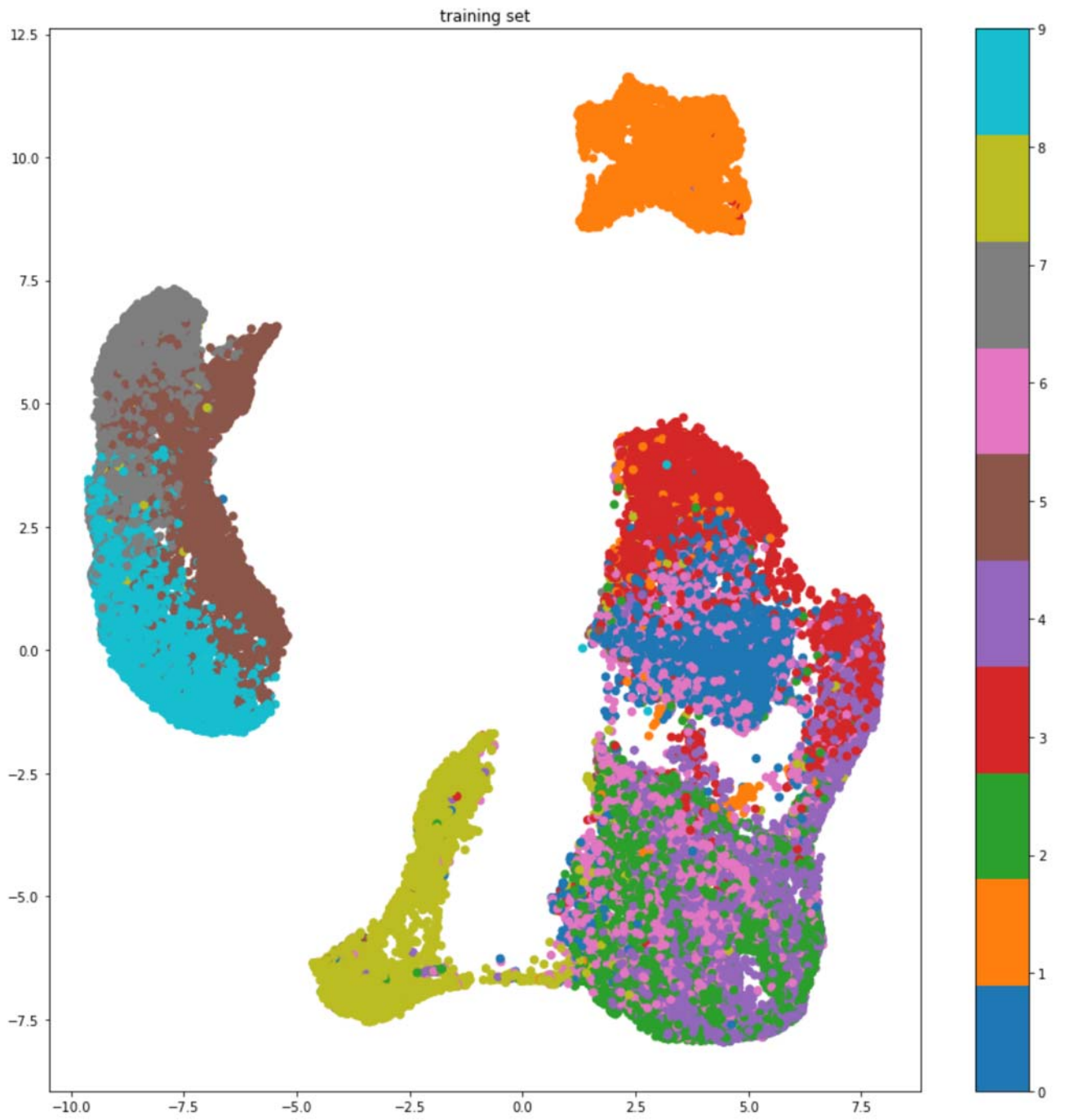
00:06:57



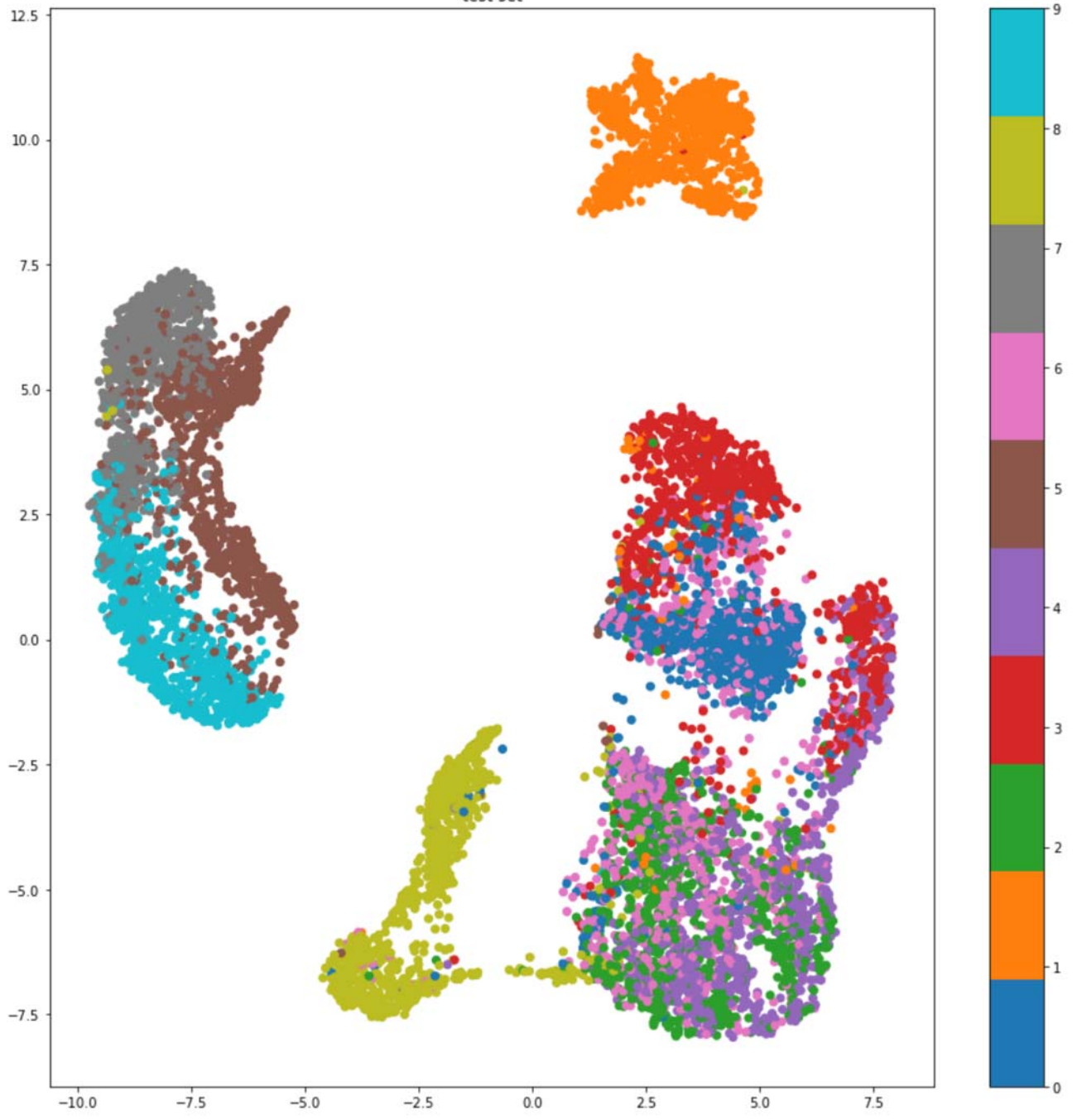
```
In [ ]: start = time.time()
umap_mod = umap.UMAP()
umap_mod.fit(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

umap_res = umap_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_train)
cbar= plt.colorbar()
plt.title('training set')
plt.show()

umap_res = umap_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_test)
cbar= plt.colorbar()
plt.title('test set')
plt.show()
```



test set



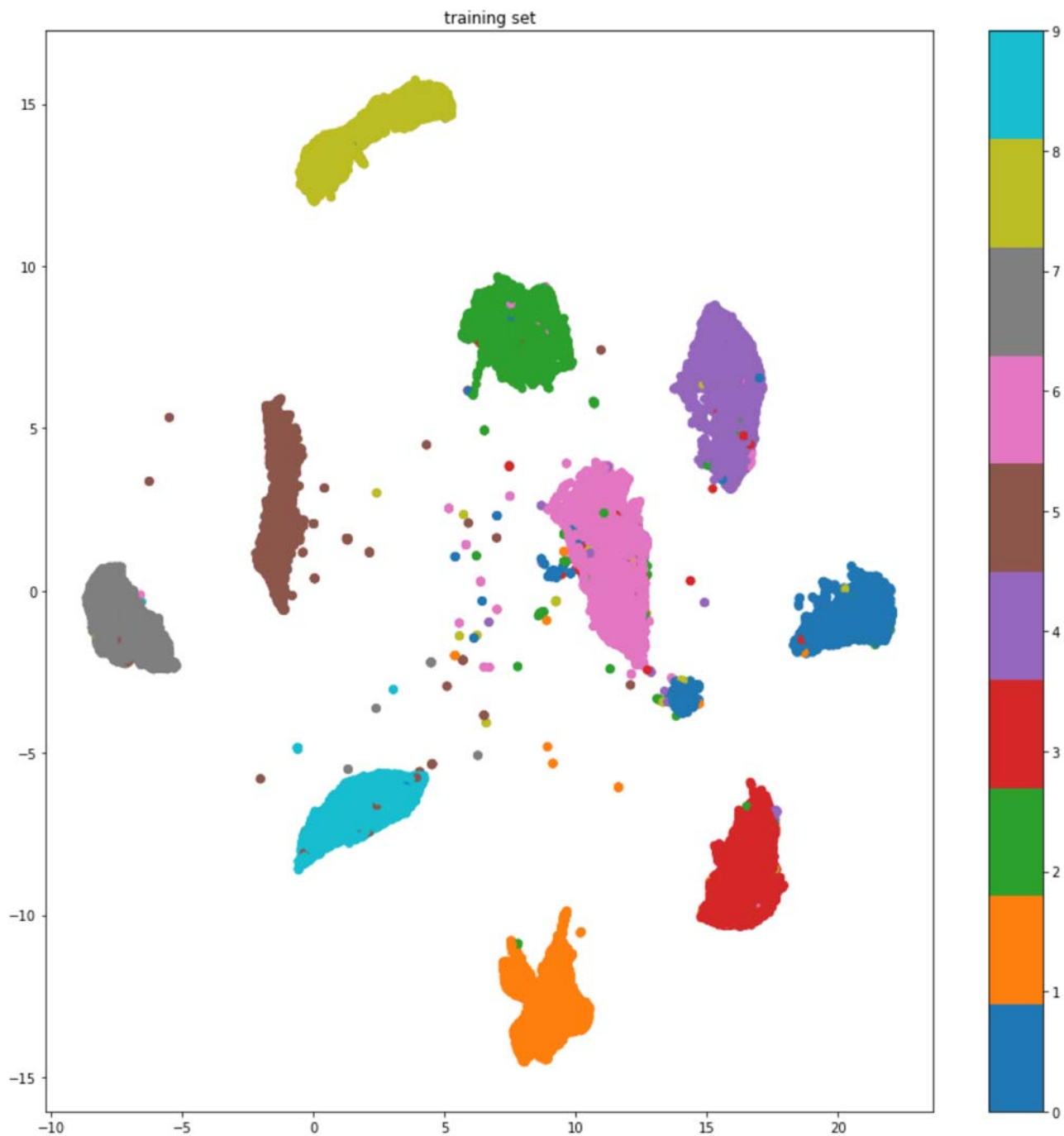
```
In [ ]: start = time.time()
        umap_mod = umap.UMAP()
        umap_mod.fit(x_train.reshape((x_train.shape[0],-1)),y_train)
        print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

        umap_res = umap_mod.transform(x_train.reshape((x_train.shape[0],-1)))
        plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_train)
        cbar= plt.colorbar()
        plt.title('training set')
        plt.show()

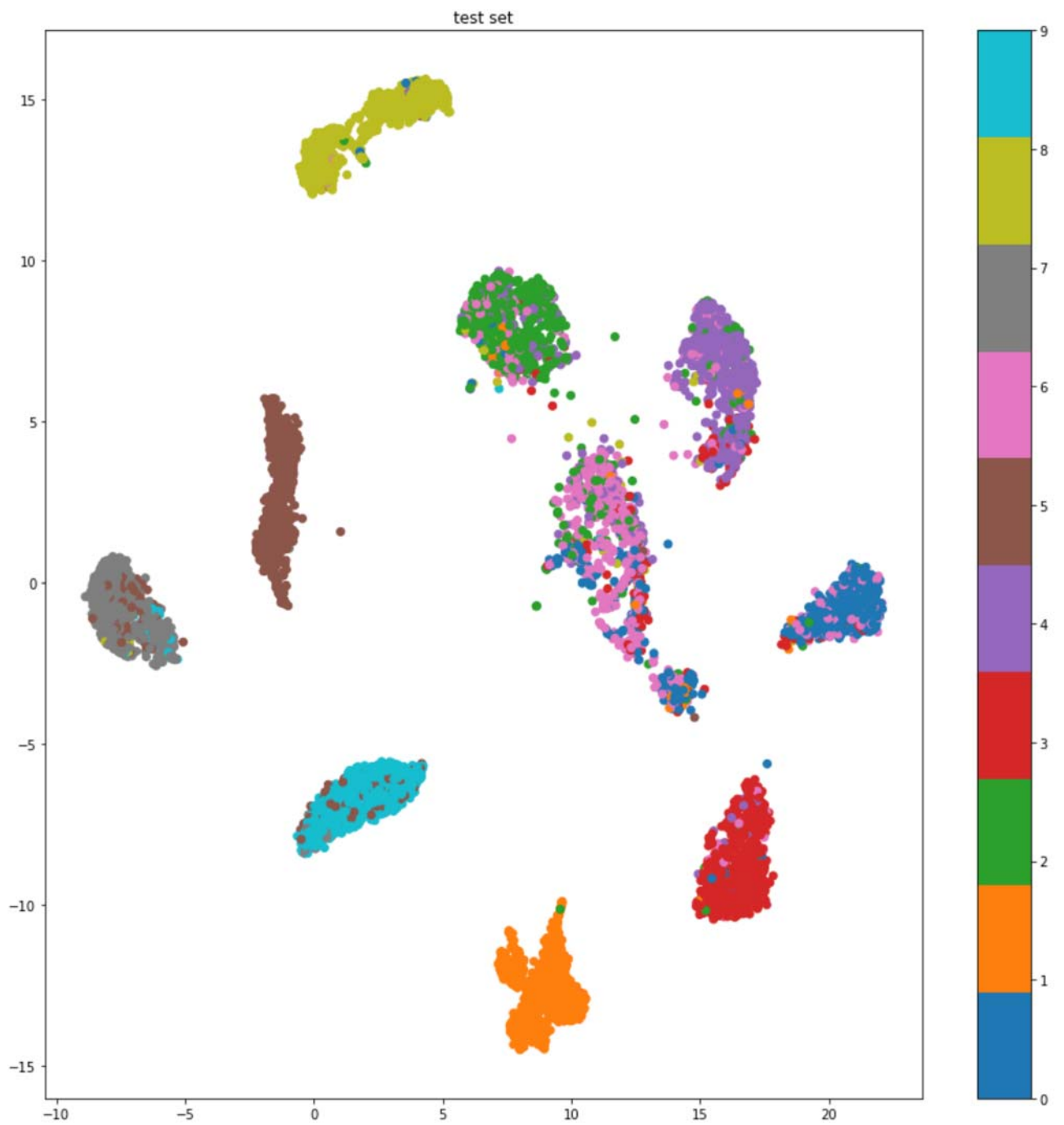
        umap_res = umap_mod.transform(x_test.reshape((x_test.shape[0],-1)))
        plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.tab10, c=y_test)
        cbar= plt.colorbar()
        plt.title('test set')
        plt.show()
```

../py3\_venv/lib/python3.5/site-packages/umap/spectral.py:229: UserWarning: Embedding a total of 4 separate connected components using meta-embedding (experimental)  
n\_components

00:02:30







```
In [3]: (x_train, y_train), (x_test, y_test) = imdb.load_data()

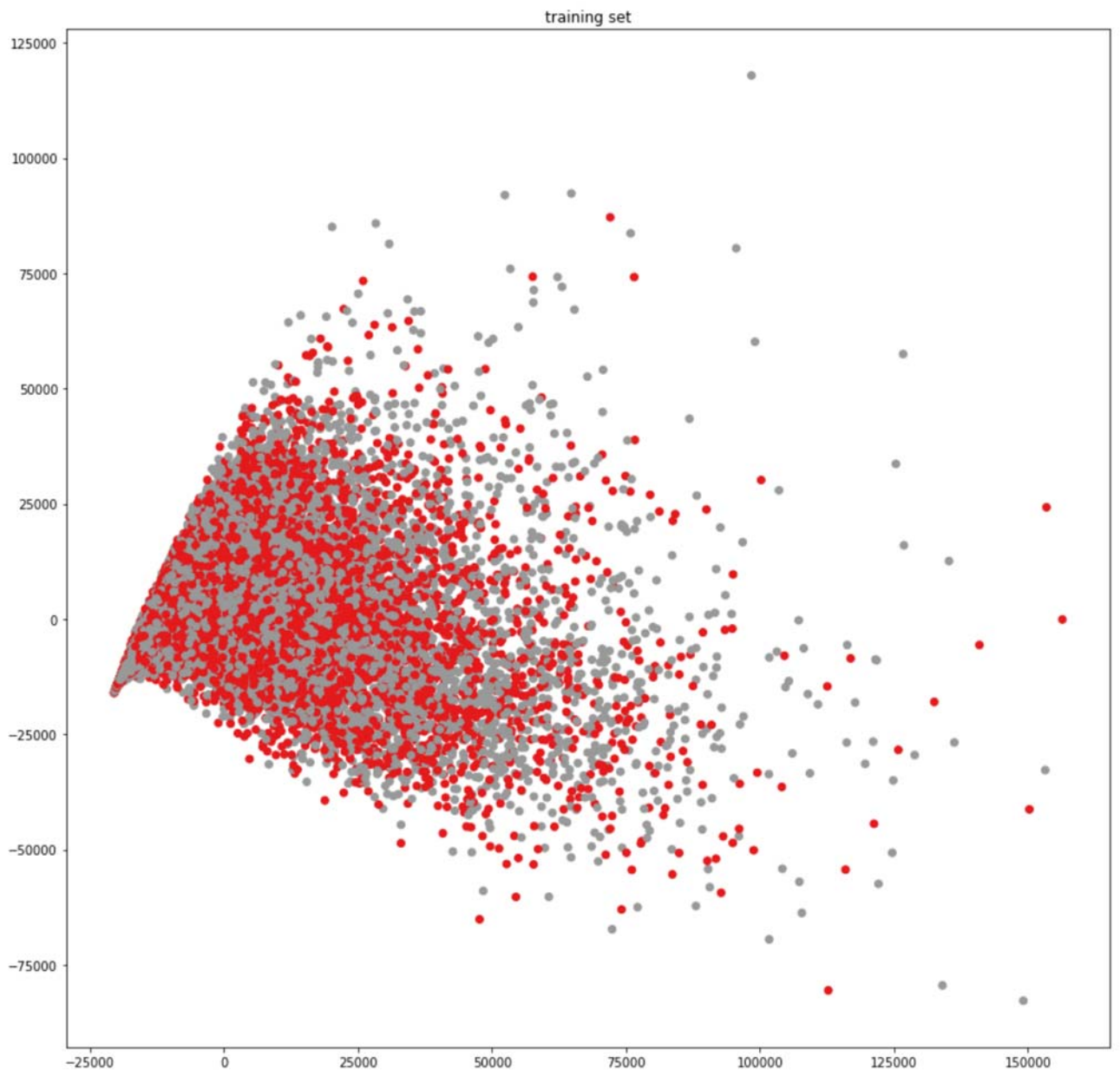
x_train, x_test = np.array(pad_sequences(x_train,maxlen=1024)), np.array(pad_sequences(x_test,maxlen=1024))
```

```
In [4]: start = time.time()
pca_mod = PCA()
pca_mod.fit(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

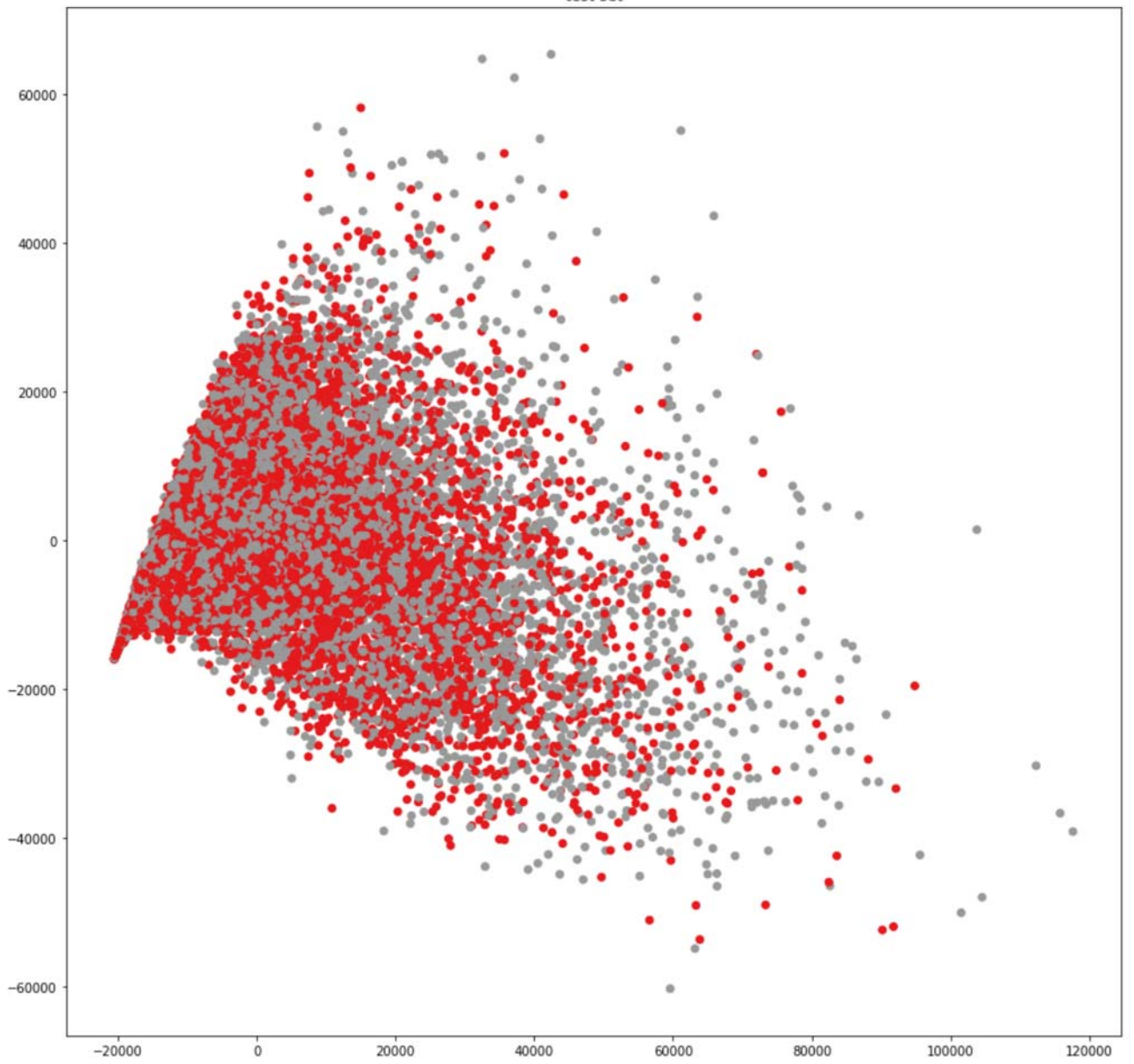
pca_res = pca_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(pca_res[:,0], pca_res[:,1], cmap=plt.cm.Set1, c=y_train)
plt.title('training set')
plt.show()

pca_res = pca_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(pca_res[:,0], pca_res[:,1], cmap=plt.cm.Set1, c=y_test)
plt.title('test set')
plt.show()
```

00:00:18



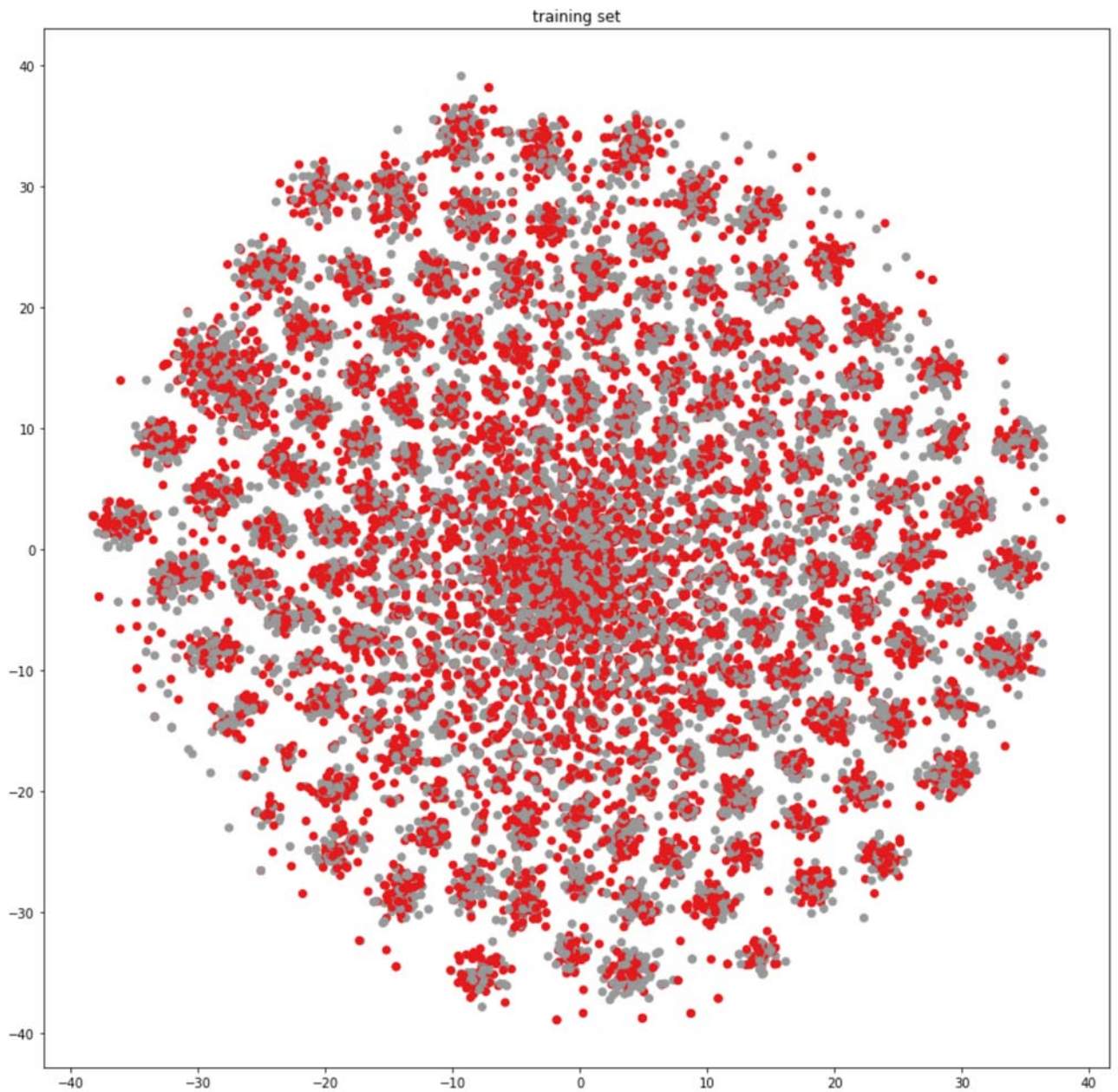
test set



```
In [5]: start = time.time()
tsne_res = TSNE(n_jobs=16).fit_transform(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

plt.scatter(tsne_res[:,0], tsne_res[:,1], cmap=plt.cm.Set1, c=y_train)
plt.title('training set')
plt.show()
```

00:04:32

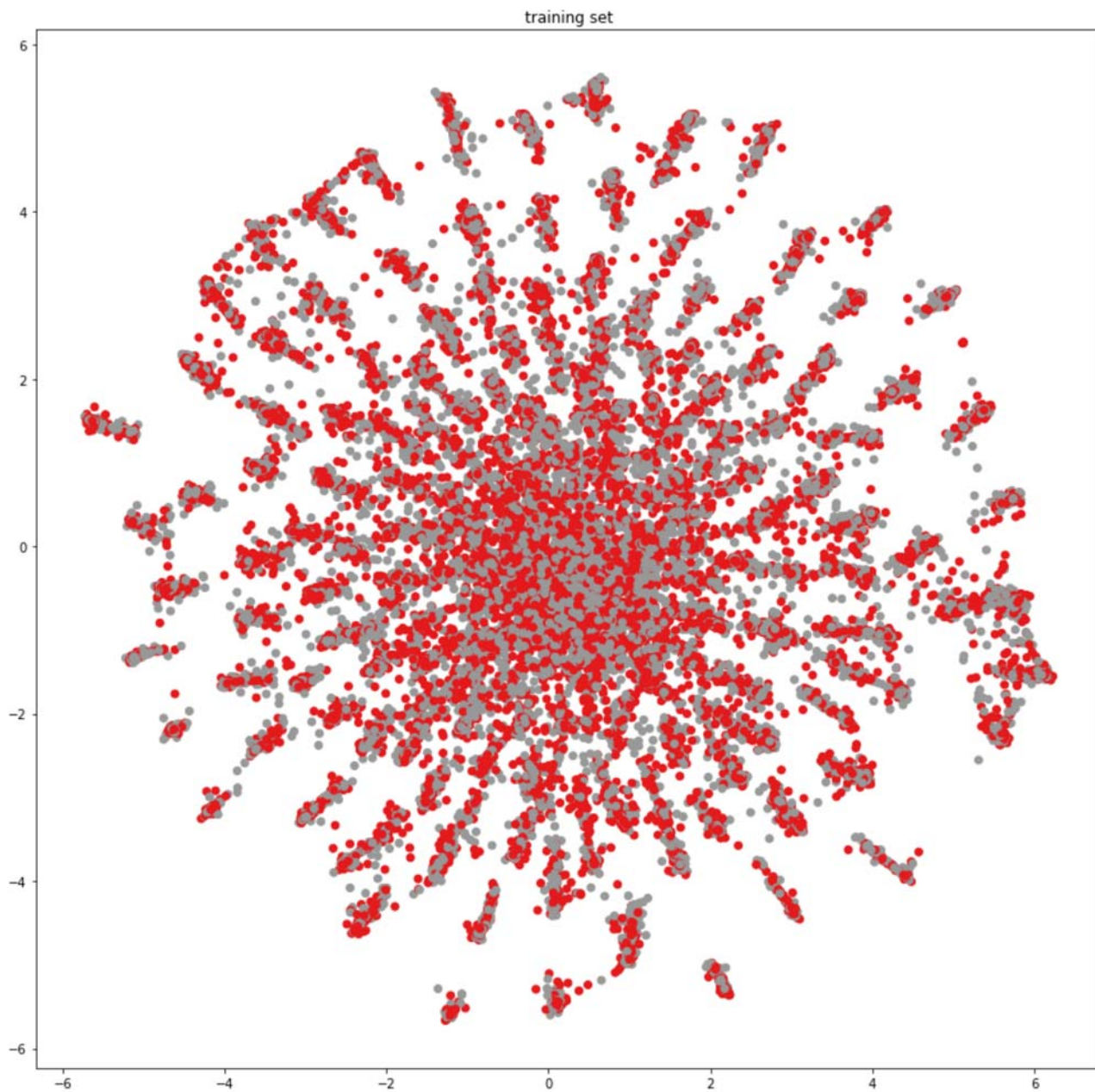


```
In [6]: start = time.time()
umap_mod = umap.UMAP()
umap_mod.fit(x_train.reshape((x_train.shape[0],-1)))
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

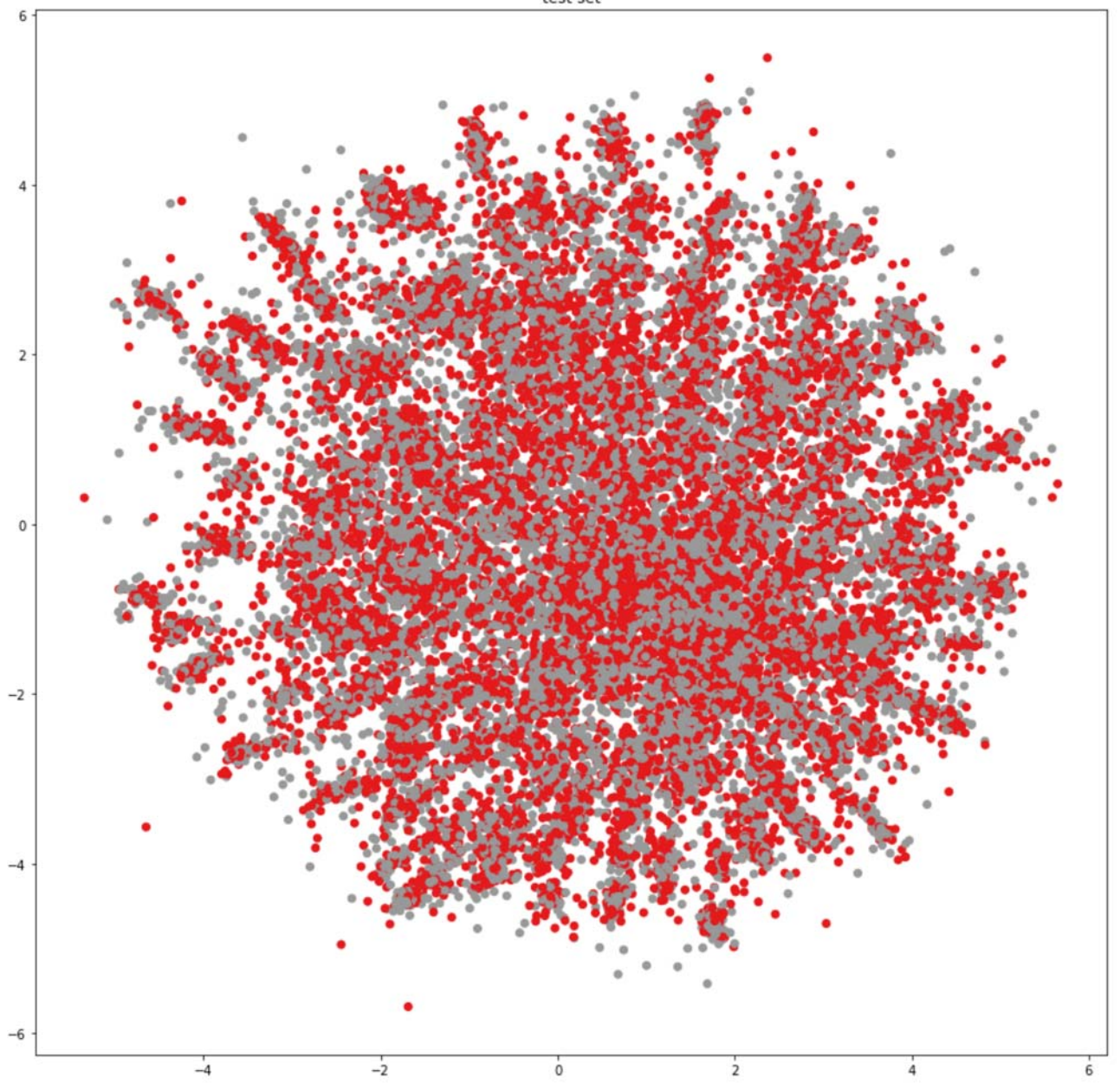
umap_res = umap_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.Set1, c=y_train)
plt.title('training set')
plt.show()

umap_res = umap_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.Set1, c=y_test)
plt.title('test set')
plt.show()
```





test set



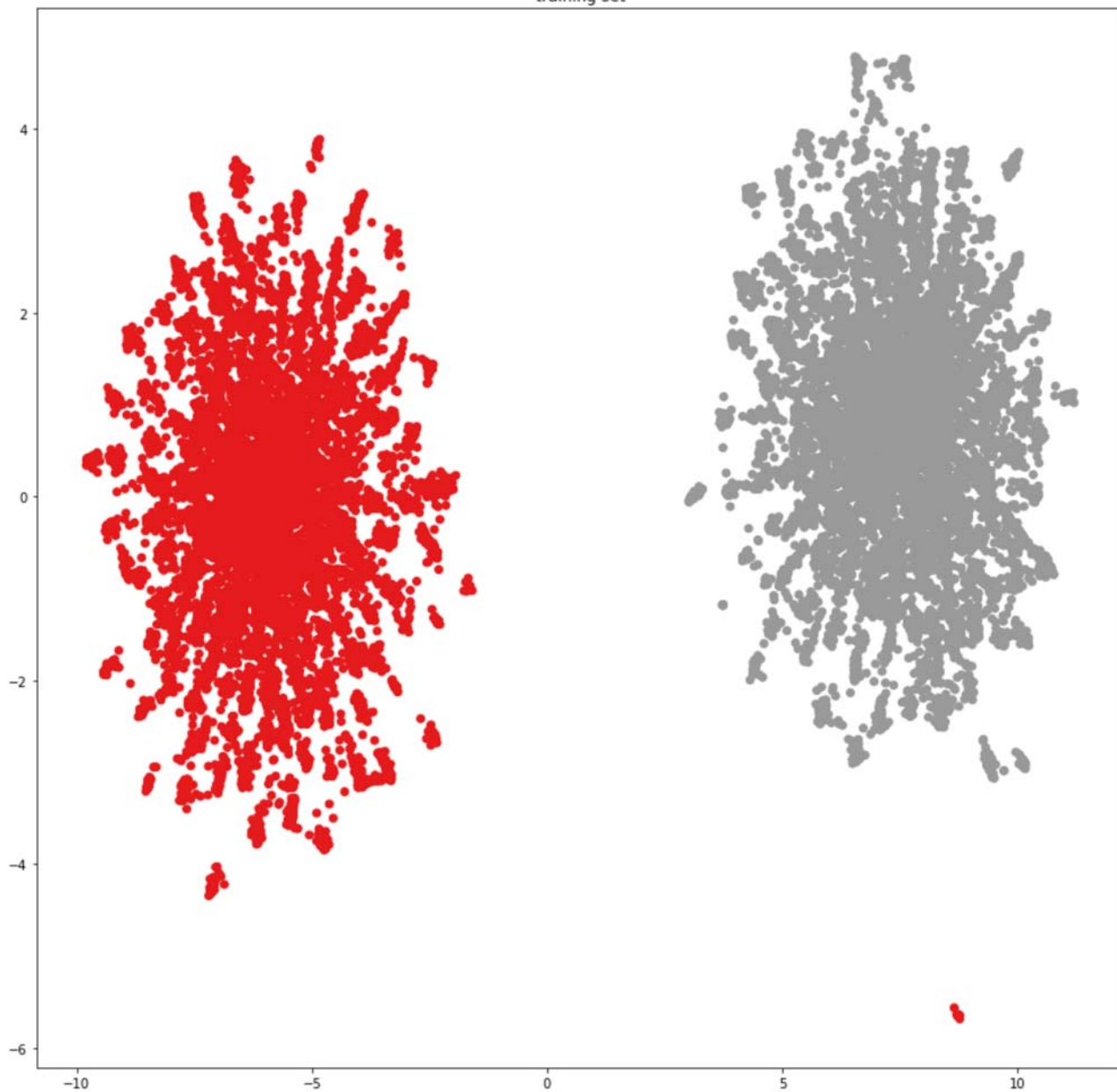
```
In [7]: start = time.time()
umap_mod = umap.UMAP()
umap_mod.fit(x_train.reshape((x_train.shape[0],-1)),y_train)
print(time.strftime('%H:%M:%S',time.gmtime(time.time()-start)))

umap_res = umap_mod.transform(x_train.reshape((x_train.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.Set1, c=y_train)
plt.title('training set')
plt.show()

umap_res = umap_mod.transform(x_test.reshape((x_test.shape[0],-1)))
plt.scatter(umap_res[:,0], umap_res[:,1], cmap=plt.cm.Set1, c=y_test)
plt.title('test set')
plt.show()
```

00:01:04

training set





test set

