In [1]:
```python
from sklearn import datasets
digits = datasets.load_digits()
dir(digits)
```

Out[1]:
```
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

In [2]:
```python
print(digits.images[0])
```

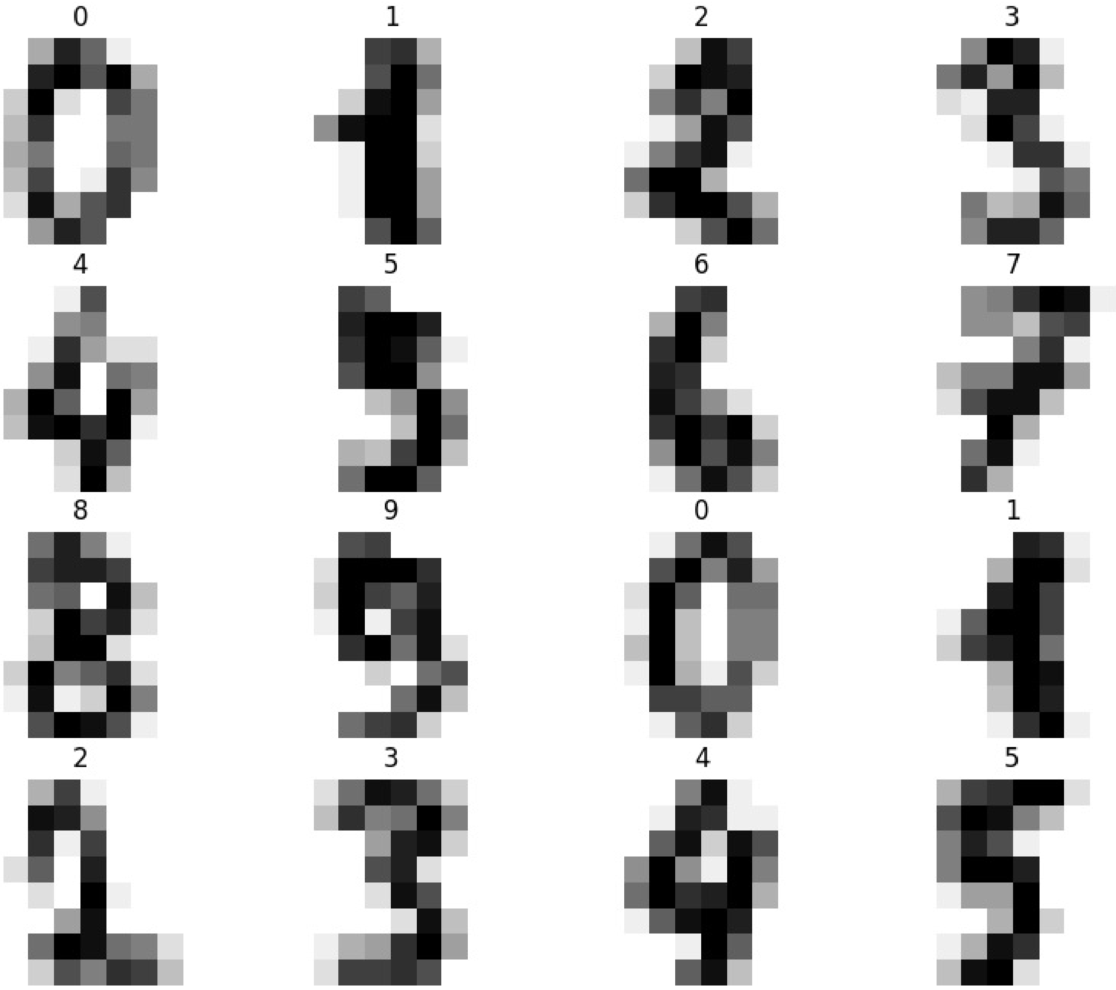
```
[[ 0.  0.  5. 13.  9.  1.  0.  0.]
 [ 0.  0. 13. 15. 10. 15.  5.  0.]
 [ 0.  3. 15.  2.  0. 11.  8.  0.]
 [ 0.  4. 12.  0.  0.  8.  8.  0.]
 [ 0.  5.  8.  0.  0.  9.  8.  0.]
 [ 0.  4. 11.  0.  1. 12.  7.  0.]
 [ 0.  2. 14.  5. 10. 12.  0.  0.]
 [ 0.  0.  6. 13. 10.  0.  0.  0.]]
```

In [6]:
```python
import matplotlib.pyplot as plt


def plot_multi(i):
        nplots = 16
        fig = plt.figure(figsize=(10, 8))
        for j in range(nplots):
                plt.subplot(4, 4, j+1)
                plt.imshow(digits.images[i+j], cmap='binary')
                plt.title(digits.target[i+j])
                plt.axis('off')

        plt.show()

plot_multi(0)
```



In [8]:
```python
y = digits.target
x = digits.images.reshape((len(digits.images), -1))

x.shape
```

Out[8]:
```
(1797, 64)
```

In [9]:
```python
x[0]
```

Out[9]:
```
array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
       15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
       12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
        0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
       10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])
```

In [10]:
```
x_train = x[:1000]
y_train = y[:1000]

x_test = x[1000:]
y_test = y[1000:]
```

In [11]:
```
from sklearn.neural_network import MLPClassifier

mlp = MLPClassifier(hidden_layer_sizes=(15,),
                    activation='logistic',
                    alpha=1e-4, solver='sgd',
                    tol=1e-4, random_state=1,
                    learning_rate_init=.1,
                    verbose=True)
```

In [12]:
```
mlp.fit(x_train, y_train)
```

```
Iteration 1, loss = 2.22958289
Iteration 2, loss = 1.91207743
Iteration 3, loss = 1.62507727
Iteration 4, loss = 1.32649842
Iteration 5, loss = 1.06100535
Iteration 6, loss = 0.83995513
Iteration 7, loss = 0.67806075
Iteration 8, loss = 0.55175832
Iteration 9, loss = 0.45840445
Iteration 10, loss = 0.39149735
Iteration 11, loss = 0.33676351
Iteration 12, loss = 0.29059880
Iteration 13, loss = 0.25437208
Iteration 14, loss = 0.22838372
Iteration 15, loss = 0.20200554
Iteration 16, loss = 0.18186565
Iteration 17, loss = 0.16461183
Iteration 18, loss = 0.14990228
Iteration 19, loss = 0.13892154
Iteration 20, loss = 0.12833784
Iteration 21, loss = 0.12138920
Iteration 22, loss = 0.11407971
Iteration 23, loss = 0.10677664
Iteration 24, loss = 0.10037149
Iteration 25, loss = 0.09593187
Iteration 26, loss = 0.09250135
Iteration 27, loss = 0.08676698
Iteration 28, loss = 0.08356043
Iteration 29, loss = 0.08209789
Iteration 30, loss = 0.07649168
Iteration 31, loss = 0.07410898
Iteration 32, loss = 0.07126869
Iteration 33, loss = 0.06926956
Iteration 34, loss = 0.06578496
Iteration 35, loss = 0.06374913
Iteration 36, loss = 0.06175492
Iteration 37, loss = 0.05975664
Iteration 38, loss = 0.05764485
Iteration 39, loss = 0.05623663
Iteration 40, loss = 0.05420966
Iteration 41, loss = 0.05413911
Iteration 42, loss = 0.05256140
Iteration 43, loss = 0.05020265
Iteration 44, loss = 0.04902779
Iteration 45, loss = 0.04788382
Iteration 46, loss = 0.04655532
Iteration 47, loss = 0.04586089
Iteration 48, loss = 0.04451758
Iteration 49, loss = 0.04341598
Iteration 50, loss = 0.04238096
Iteration 51, loss = 0.04162200
Iteration 52, loss = 0.04076839
Iteration 53, loss = 0.04003180
Iteration 54, loss = 0.03907774
Iteration 55, loss = 0.03815565
Iteration 56, loss = 0.03791975
Iteration 57, loss = 0.03706276
Iteration 58, loss = 0.03617874
Iteration 59, loss = 0.03593227
Iteration 60, loss = 0.03504175
```

```
Iteration 61, loss = 0.03441259
Iteration 62, loss = 0.03397449
Iteration 63, loss = 0.03326990
Iteration 64, loss = 0.03305025
Iteration 65, loss = 0.03244893
Iteration 66, loss = 0.03191504
Iteration 67, loss = 0.03132169
Iteration 68, loss = 0.03079707
Iteration 69, loss = 0.03044946
Iteration 70, loss = 0.03005546
Iteration 71, loss = 0.02960555
Iteration 72, loss = 0.02912799
Iteration 73, loss = 0.02859103
Iteration 74, loss = 0.02825959
Iteration 75, loss = 0.02788968
Iteration 76, loss = 0.02748725
Iteration 77, loss = 0.02721247
Iteration 78, loss = 0.02686225
Iteration 79, loss = 0.02635636
Iteration 80, loss = 0.02607439
Iteration 81, loss = 0.02577613
Iteration 82, loss = 0.02553642
Iteration 83, loss = 0.02518749
Iteration 84, loss = 0.02484300
Iteration 85, loss = 0.02455379
Iteration 86, loss = 0.02432480
Iteration 87, loss = 0.02398548
Iteration 88, loss = 0.02376004
Iteration 89, loss = 0.02341261
Iteration 90, loss = 0.02318255
Iteration 91, loss = 0.02296065
Iteration 92, loss = 0.02274048
Iteration 93, loss = 0.02241054
Iteration 94, loss = 0.02208181
Iteration 95, loss = 0.02190861
Iteration 96, loss = 0.02174404
Iteration 97, loss = 0.02156939
Iteration 98, loss = 0.02119768
Iteration 99, loss = 0.02101874
Iteration 100, loss = 0.02078230
Iteration 101, loss = 0.02061573
Iteration 102, loss = 0.02039802
Iteration 103, loss = 0.02017245
Iteration 104, loss = 0.01997162
Iteration 105, loss = 0.01989280
Iteration 106, loss = 0.01963828
Iteration 107, loss = 0.01941850
Iteration 108, loss = 0.01933154
Iteration 109, loss = 0.01911473
Iteration 110, loss = 0.01905371
Iteration 111, loss = 0.01876085
Iteration 112, loss = 0.01860656
Iteration 113, loss = 0.01848655
Iteration 114, loss = 0.01834844
Iteration 115, loss = 0.01818981
Iteration 116, loss = 0.01798523
Iteration 117, loss = 0.01783630
Iteration 118, loss = 0.01771441
Iteration 119, loss = 0.01749814
Iteration 120, loss = 0.01738339
Iteration 121, loss = 0.01726549
Iteration 122, loss = 0.01709638
Iteration 123, loss = 0.01698340
Iteration 124, loss = 0.01684606
Iteration 125, loss = 0.01667016
Iteration 126, loss = 0.01654172
Iteration 127, loss = 0.01641832
Iteration 128, loss = 0.01630111
Iteration 129, loss = 0.01623051
Iteration 130, loss = 0.01612736
Iteration 131, loss = 0.01590220
Iteration 132, loss = 0.01582485
Iteration 133, loss = 0.01571372
Iteration 134, loss = 0.01560349
Iteration 135, loss = 0.01557688
Iteration 136, loss = 0.01534420
Iteration 137, loss = 0.01527883
Iteration 138, loss = 0.01517545
Iteration 139, loss = 0.01503663
Iteration 140, loss = 0.01501192
Iteration 141, loss = 0.01482535
Iteration 142, loss = 0.01471388
Iteration 143, loss = 0.01463948
```

```
Iteration 144, loss = 0.01454059
Iteration 145, loss = 0.01441742
Iteration 146, loss = 0.01431741
Iteration 147, loss = 0.01428414
Iteration 148, loss = 0.01416364
Iteration 149, loss = 0.01406742
Iteration 150, loss = 0.01402651
Iteration 151, loss = 0.01389720
Iteration 152, loss = 0.01381412
Iteration 153, loss = 0.01371300
Iteration 154, loss = 0.01362465
Iteration 155, loss = 0.01357048
Iteration 156, loss = 0.01348760
Iteration 157, loss = 0.01339543
Iteration 158, loss = 0.01331941
Iteration 159, loss = 0.01320812
Iteration 160, loss = 0.01315415
Iteration 161, loss = 0.01308279
Iteration 162, loss = 0.01302708
Iteration 163, loss = 0.01290042
Iteration 164, loss = 0.01289267
Iteration 165, loss = 0.01277558
Iteration 166, loss = 0.01277238
Iteration 167, loss = 0.01261308
Iteration 168, loss = 0.01260611
Iteration 169, loss = 0.01248789
Iteration 170, loss = 0.01239662
Iteration 171, loss = 0.01231743
Iteration 172, loss = 0.01227346
Iteration 173, loss = 0.01223136
Iteration 174, loss = 0.01217211
Iteration 175, loss = 0.01208682
Iteration 176, loss = 0.01204707
Iteration 177, loss = 0.01200225
Iteration 178, loss = 0.01188677
Iteration 179, loss = 0.01184993
Iteration 180, loss = 0.01175130
Iteration 181, loss = 0.01171178
Iteration 182, loss = 0.01166052
Iteration 183, loss = 0.01163843
Iteration 184, loss = 0.01154892
Iteration 185, loss = 0.01147629
Iteration 186, loss = 0.01142365
Iteration 187, loss = 0.01136608
Iteration 188, loss = 0.01128053
Iteration 189, loss = 0.01128869
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs. Stopping.
```

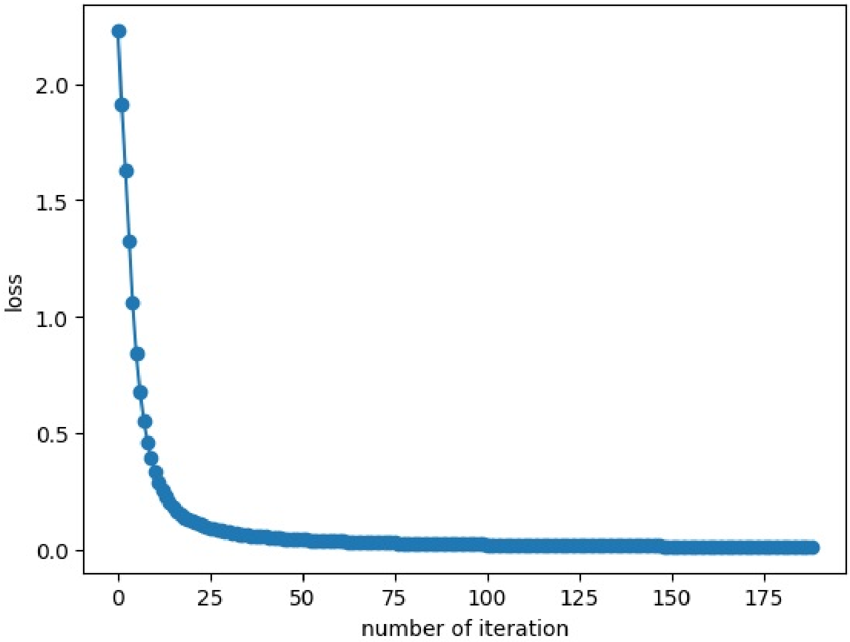
Out[12]:

MLPClassifier

```
MLPClassifier(activation='logistic', hidden_layer_sizes=(15,),
              learning_rate_init=0.1, random_state=1, solver='sgd',
              verbose=True)
```

In [13]:

```python
fig, axes = plt.subplots(1, 1)
axes.plot(mlp.loss_curve_, 'o-')
axes.set_xlabel("number of iteration")
axes.set_ylabel("loss")
plt.show()
```



In [14]:
```
predictions = mlp.predict(x_test)
predictions[:50]
```

Out[14]:
```
array([1, 4, 0, 5, 3, 6, 9, 6, 1, 7, 5, 4, 4, 7, 2, 8, 2, 2, 5, 7, 9, 5,
       4, 4, 9, 0, 8, 9, 8, 0, 1, 2, 3, 4, 5, 6, 7, 8, 3, 0, 1, 2, 3, 4,
       5, 6, 7, 8, 5, 0])
```

In [15]:
```
y_test[:50]
```

Out[15]:
```
array([1, 4, 0, 5, 3, 6, 9, 6, 1, 7, 5, 4, 4, 7, 2, 8, 2, 2, 5, 7, 9, 5,
       4, 4, 9, 0, 8, 9, 8, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4,
       5, 6, 7, 8, 9, 0])
```

In [16]:
```
from sklearn.metrics import accuracy_score

accuracy_score(y_test, predictions)
```

Out[16]:
```
0.9146800501882058
```

In [ ]: