|  |  |
| --- | --- |
| 11. DIMENSIONALLY REDUCTION ALGORITHM | |
| IN [1] | **import** numpy **as** np  **import** pandas **as** pd  **import** os |
| IN [2] | **import** matplotlib.pyplot **as** plt  **from** sklearn.model\_selection **import** train\_test\_split  **from** sklearn.neural\_network **import** MLPClassifier  **from** sklearn.metrics **import** accuracy\_score, confusion\_matrix  **from** sklearn.decomposition **import** PCA  X **=** np**.**load(r"C:\Users\malar\Downloads\Compressed\X.npy\X.npy")  Y **=** np**.**load(r"C:\Users\malar\Downloads\Y.npy")  X**.**shape |
| IN [3] | plt**.**imshow(X[0]) |
| IN [4] | 9 **-** np**.**argmax(Y[0]) |
| IN [5] | X\_flat **=** np**.**array(X)**.**reshape((2062, 64**\***64))  X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X\_flat, Y, test\_size**=**0.3, random\_state**=**42) |
| IN [6] | clf **=** MLPClassifier(solver**=**'adam', alpha**=**1e-5, hidden\_layer\_sizes**=**(20, 20, 20), random\_state**=**1)  clf**.**fit(X\_train, y\_train) |
| IN [7] | y\_hat **=** clf**.**predict(X\_test)  print("accuracy: " **+** str(accuracy\_score(y\_test, y\_hat))) |
| IN [8] | pca\_dims **=** PCA()  pca\_dims**.**fit(X\_train)  cumsum **=** np**.**cumsum(pca\_dims**.**explained\_variance\_ratio\_)  d **=** np**.**argmax(cumsum **>=** 0.95) **+** 1 |
| IN [9] | d |
| IN [10] | pca **=** PCA(n\_components**=**d)  X\_reduced **=** pca**.**fit\_transform(X\_train)  X\_recovered **=** pca**.**inverse\_transform(X\_reduced) |
| IN [11] | print("reduced shape: " **+** str(X\_reduced**.**shape))  print("recovered shape: " **+** str(X\_recovered**.**shape)) |
| IN [12] | f **=** plt**.**figure()  f**.**add\_subplot(1,2, 1)  plt**.**title("original")  plt**.**imshow(X\_train[0]**.**reshape((64,64)))  f**.**add\_subplot(1,2, 2)  plt**.**title("PCA compressed")  plt**.**imshow(X\_recovered[0]**.**reshape((64,64)))  plt**.**show(block**=True**) |
| IN [13] | clf\_reduced **=** MLPClassifier(solver**=**'adam', alpha**=**1e-5, hidden\_layer\_sizes**=**(20, 20, 20))  clf\_reduced**.**fit(X\_reduced, y\_train) |
| IN [14] | X\_test\_reduced **=** pca**.**transform(X\_test)  y\_hat\_reduced **=** clf\_reduced**.**predict(X\_test\_reduced)  print("accuracy: " **+** str(accuracy\_score(y\_test, y\_hat\_reduced))) |

|  |  |
| --- | --- |
| OUT [2] | (2062, 64, 64) |
| OUT [3] |  |
| OUT [7] | accuracy: 0.3360258481421648 |
| OUT [9] | 292 |
| OUT [11] |  |
| OUT [14] | accuracy: 0.6155088852988692 |