CSE 546: MACHINE LEARNING HOMEWORK 3

DAVID P. FLEMING

Contents

Introduction

Please note that a copy of all the code I wrote to answer the questions in this assignment are included in my submission but also located online at `https://github.com/dflemin3/CSE_546/tree/master/HW3`. Some scripts require data, such as MNIST data, to run to completion and were not included on my github due to file size constraints. The MNIST data is included in the `Data` directory as python `.pkl` files as this compressed format gave me quicker load times for my scripts.

Overall, my code is structured as follows: There are three main directories included in my submission: `DML`, `Data` and `HW3`. `HW3` contains all the scripts used to run my analysis. For example to reproduce the answer for Question 1.2, one would run `python hw3_1.2.py`. All scripts have relative file paths so the code should run and have detailed comments to describe functionality. In addition, the scripts have flags near the top to control functionality. By default, I set all flags to true so the script performs the entire analysis and plotting.

The `Data` directory contains both the MNIST dataset. The grader should be able to run my homework scripts without altering this directory.

The `DML` directory contains all the auxiliary files used to do the computations in the homework scripts and has a logical hierarchy. For example, the directory `optimization` contains the file `gradient_descent.py` while contains both my batch gradient descent and stochastic gradient descent implementations.

Date: November xxxth, 2016.

The directory `data_processing` contains the script `mnist_utils.py` which contains my functions used to load and work with the MNIST data. The directory `classification` contains the file `classifier_utils.py` which contains all things related to binary and softmax classification including the gradients for each respective method for use with a gradient descent algorithm. The `validation` subdirectory contains `validation.py`. This file contains all my loss functions such as 0/1 loss and also my implementations for regularization paths for both linear regression and logistic and softmax classification using gradient descent. Finally, the `regression` directory contains all utilities for a normal or multi-class regression. In particular, this directory contains the file where my ridge regression implementation lives, `ridge_utils.py`.

In each section, I try to be explicit with what files I used to perform the computation including the path from the `DML` directory for ease of grading.

Question 0: Collaborators

I collaborated with Matt Wilde, Serena Liu, and Janet Matsen for various questions on this assignment.

Question 1: PCA and reconstruction

TODO

In this question, I solve 10 linear regression problems using regularized Ridge Regression for all 10 digits of the MNIST dataset to build a "one vs all" classifier. Each regression is a binary classifier for the corresponding digit. I classify a sample according to the largest predicted score among my 10 predictors. The code used to solve this question is in the following attached files: `hw3_1.2.py, hw3_1.3.py, hw3_1.4.py` in the `HW3` directory and `classification/classifier_utils.py, regression/regression_utils.py, validation/validation.py, data_processing/mnist_` in the `DML` directory.

1.1: Matrix Algebra Review. TODO

1.2: PCA.

1.2.1. TODO

1.2.2. TODO

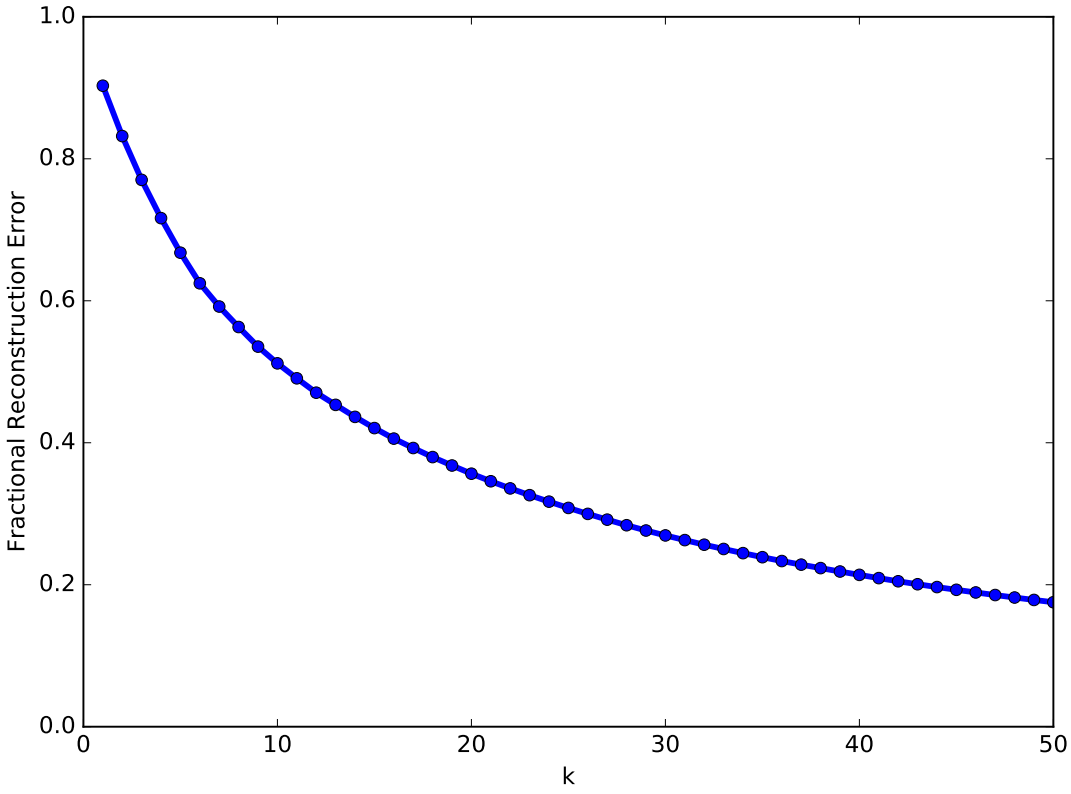

FIGURE 1. Fractional reconstruction error as a function of k principal components out of d total dimensions for $k \in [1, 50]$.

1.2.3. TODO

1.2.4. TODO

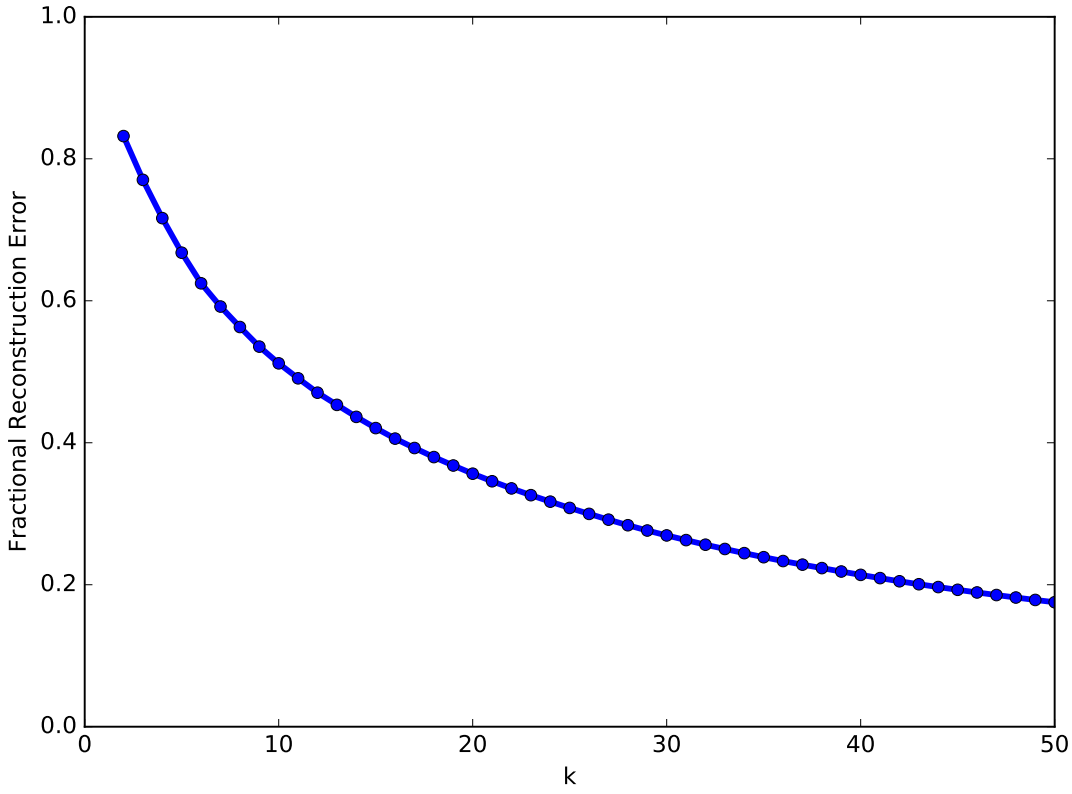

FIGURE 2. Fractional reconstruction error as a function of k principal components out of d total dimensions for $k \in [2, 50]$.

1.3: Visualization of the Eigen-Directions.

1.3.1. TODO

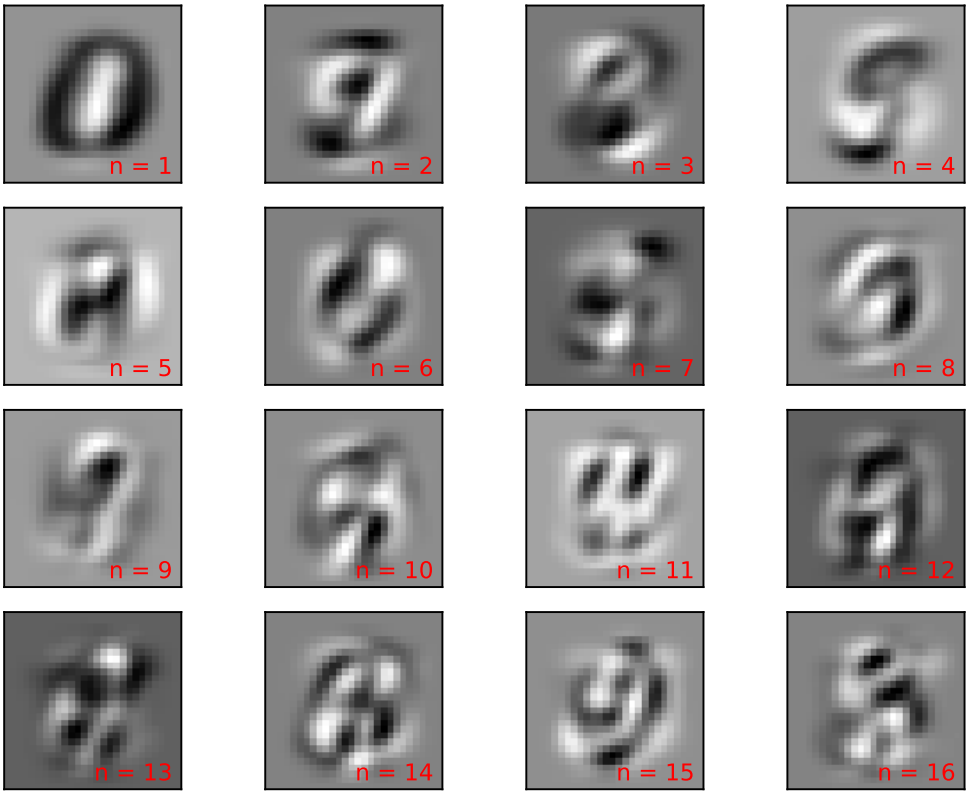

FIGURE 3. Visualization of the first 16 eigendirections as 28×28 pixel images. The respective eigendirection number is given in red in a subplot's lower righthand corner.

1.3.2. TODO: eigen-interpretations

1.4: Visualization and Reconstruction. TODO

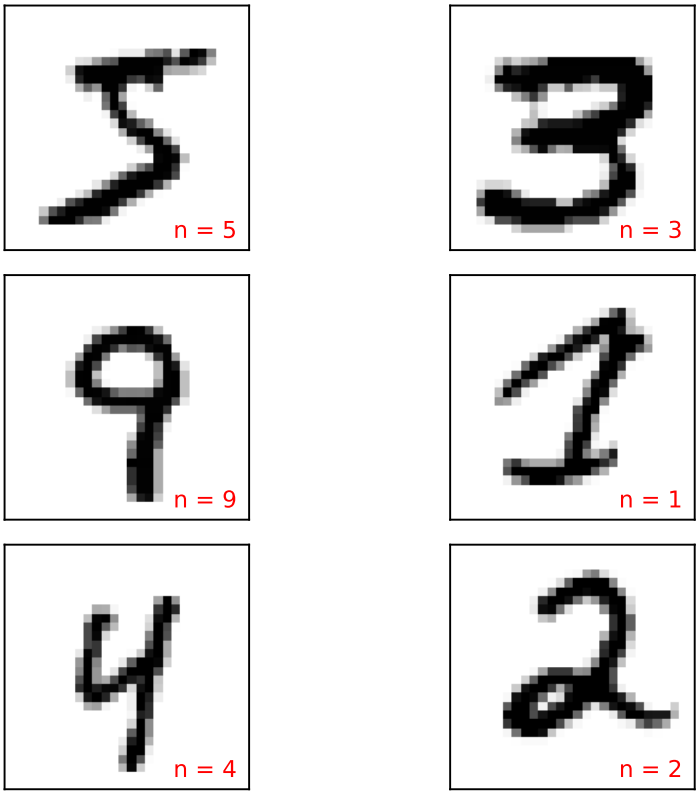

FIGURE 4. Visualization of 6 random, unique digits from the MNIST training set. The image label is displayed in red in the lower-righthand corner of each image.

1.4.1.

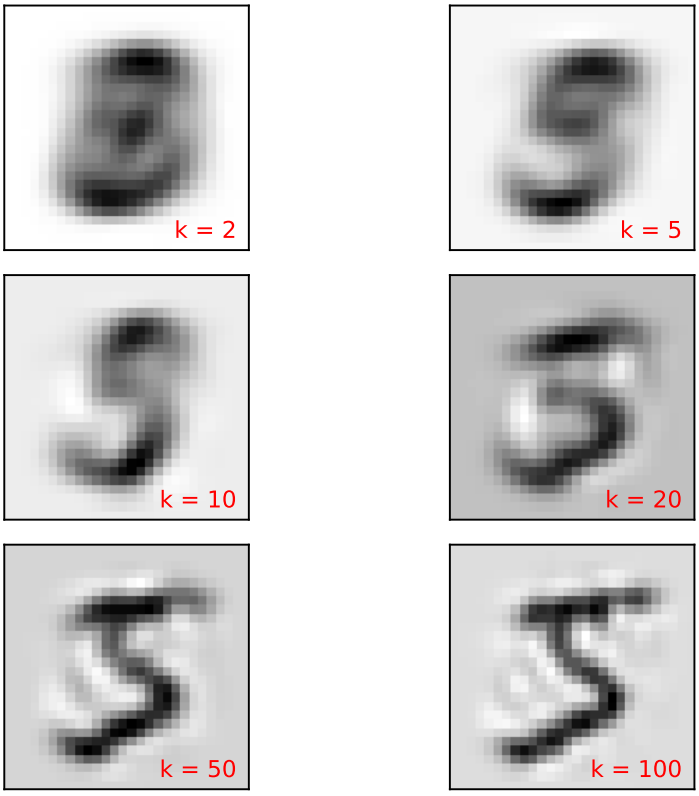

FIGURE 5. Reconstruction of a number 5 from the MNIST data set (see Fig. 4 using $k \in [2, 5, 10, 20, 50, 100]$ principal components where the given k is denoted in red in the lower-righthand corner of the respective subplot.

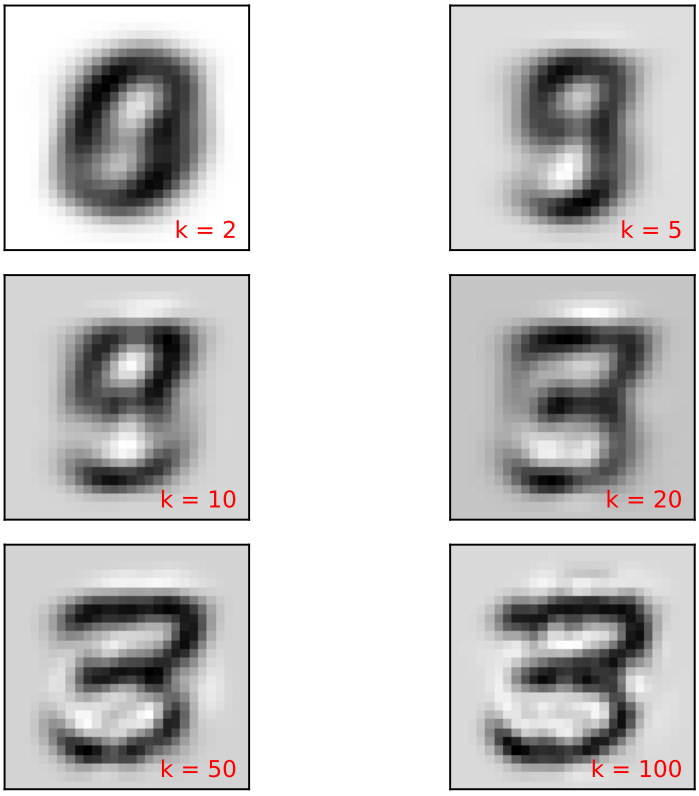

FIGURE 6. Reconstruction of a number 3 from the MNIST data set (see Fig. 4) using $k \in [2, 5, 10, 20, 50, 100]$ principal components where the given k is denoted in red in the lower-righthand corner of the respective subplot.

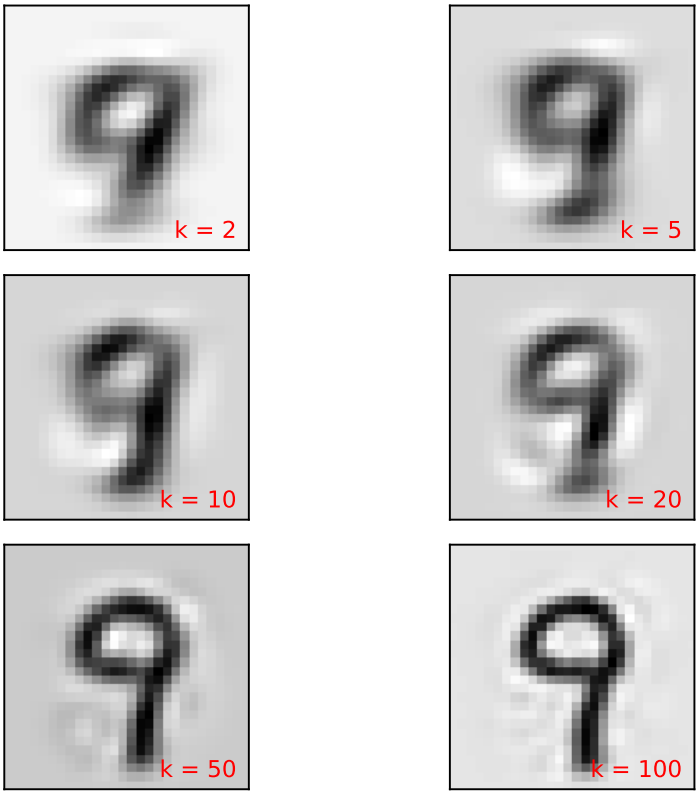

FIGURE 7. Reconstruction of a number 9 from the MNIST data set (see Fig. 4) using $k \in [2, 5, 10, 20, 50, 100]$ principal components where the given k is denoted in red in the lower-righthand corner of the respective subplot.

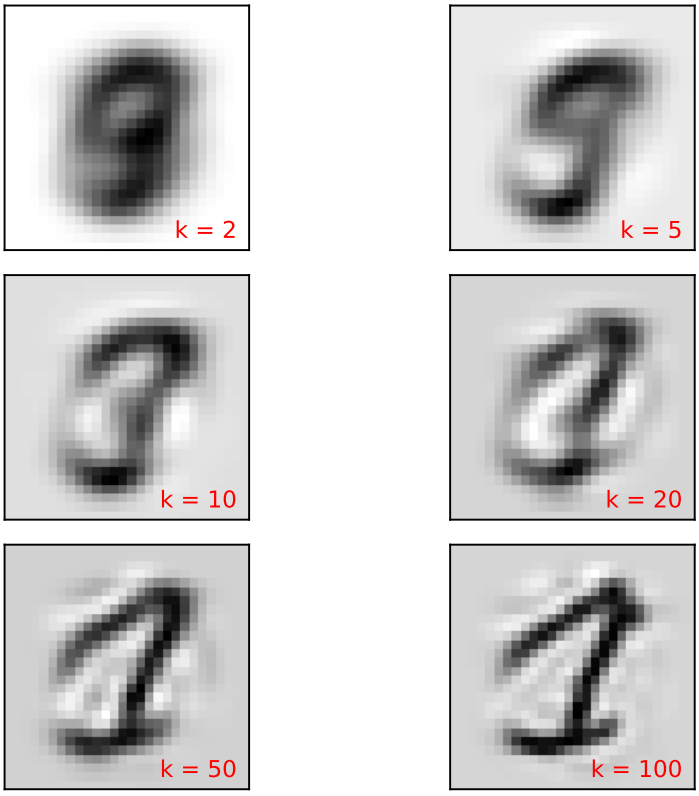

FIGURE 8. Reconstruction of a number 1 from the MNIST data set (see Fig. 4) using $k \in [2, 5, 10, 20, 50, 100]$ principal components where the given k is denoted in red in the lower-righthand corner of the respective subplot.

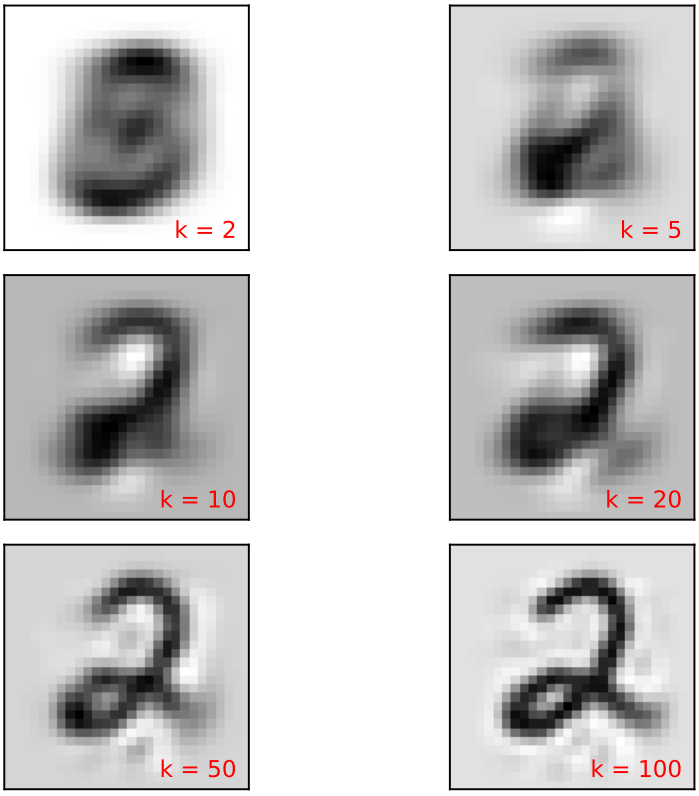

FIGURE 9. Reconstruction of a number 2 from the MNIST data set (see Fig. 4) using $k \in [2, 5, 10, 20, 50, 100]$ principal components where the given k is denoted in red in the lower-righthand corner of the respective subplot.

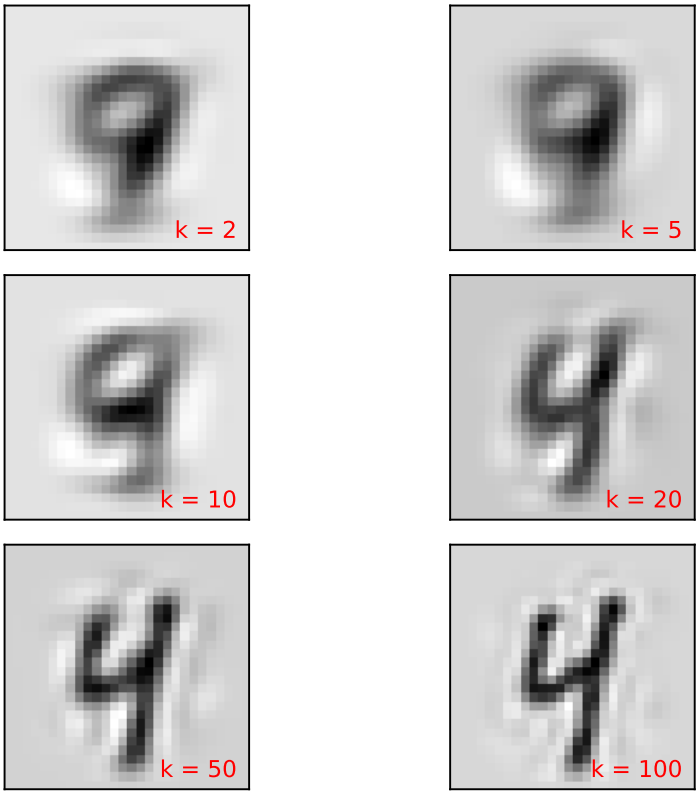

FIGURE 10. Reconstruction of a number 4 from the MNIST data set (see Fig. 4) using $k \in [2, 5, 10, 20, 50, 100]$ principal components where the given k is denoted in red in the lower-righthand corner of the respective subplot.

1.4.2.

1.4.3. TODO