In this part, you are going to play with the "handwritten digit recognition" problem on the MNIST data set. The task is to classify handwritten images of numbers between 0 to 9.

You are **NOT** allowed to use any of the pre-built classifiers in **sklearn**. Feel free to use any method from **numpy** or **scipy**.

Get the data from https://pypi.python.org/pypi/python-mnist. Load the data as following figure:

```
from mnist import MNIST

def load_dataset():
    mndata = MNIST('./data/')
    X_train, labels_train = map(np.array, mndata.load_training())
    X_test, labels_test = map(np.array, mndata.load_testing())
    X_train = X_train/255.0
    X_test = X_test/255.0
```

Or you can also get data from Tenserflow as following:

```
import tensorflow as tf
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
```


Each example has features $x_i \in \mathrm{R}^d$ (with $d = 28*28 = 784$) and label $z_j \in \{0,...,9\}$. You can visualize a single example x_i with *imshow* after reshaping it to its original 28 x 28 image shape. We wish to learn a classifier that takes as input a vector in R^d and outputs an index in $\{0,...,9\}$.

1. *(20 points)* Use the gradient descent algorithm to train a multi-class logistic regression classifier. Plot (1) the objective value (log-likelihood), (2) the training accuracy, and (3) the testing accuracy versus the number of iterations. Report your final testing accuracy, i.e. the fraction of test images that are correctly classified.

   Note that you must choose a suitable learning rate (i.e. step size) of the gradient descent algorithm. A hint is that your learning rate cannot be too large otherwise your objective will increase only for the first few iterations.

   In addition, you need to choose a suitable stopping criterion. You might use the number of iterations, the decrease of the objective value, or the maximum of the L_2 norms of the gradient with respect to each w_k. Or you might watch the increase of the testing accuracy and stop the optimization when the accuracy is stable.

2. *(20 points)* Now we add the regularization term $\frac{\lambda}{2}\sum_{l=1}^{K-1}||w_l||_2^2$. For λ = 1, 10, 100, 1000, report the final testing accuracies.

3. *(10 points)* What can you conclude from the above experiment? (hint: the relationship between the regularization weight and the prediction performance).

Submission Instructions

To earn the full credit, your **must type** your solutions with sufficient details either using LaTeX or Microsoft Word and submit it through the blackboard website. You will get points off for not following this requirement.