NISARG UPADHYAYA - 19CS30031

Assignment 2 Q10

Link to colab:
https://colab.research.google.com/drive/1Qy3sUp7AaSQEXnFwOyLlrUuwNpEuxXdZ?usp=sharing

```
Training examples = (10000, 784), (10000,)
Testing examples = (1000, 784), (1000,)
```

Training

```
Confusion Matrix:
[[951   1   3   2   6   8  15   0  13   1]
 [  0 971   3   2   2   3   1   0  17   1]
 [ 16  28 841  24  14   0  30   9  37   1]
 [  5  19  21 877   0  18   6   9  25  20]
 [  3  14   5   1 893  10   6   3  14  51]
 [ 14   9   9  62   8 760  23   9  84  22]
 [ 15   7  13   2   8  12 930   0  13   0]
 [  4  26   6   6  22   1   0 867   6  62]
 [ 12  67   7  34  21  38   7   8 777  29]
 [ 11  13   4  17  55   2   0  61   8 829]]

Accuracy: 86.96%
```

Test

```
Confusion Matrix:
[[ 79   0   0   1   0   3   2   0   1   1]
 [  0 115   0   0   0   0   0   0   6   0]
 [  1   5  78   5   6   0   1   0   4   0]
 [  0   1   0  77   0   2   1   0   3   1]
 [  0   4   0   0 105   1   1   1   2   5]
 [  2   1   2  12   1  58   1   1   7   6]
 [  1   0   2   0   2   1  97   0   0   0]
 [  1   4   2   3   3   1   0  76   0  13]
 [  3   6   1   0   1   4   0   3  78   2]
 [  1   1   0   1  11   0   1   9   1  68]]

Accuracy: 83.1%
```

CODE

```
def get_data():
   (x_train, y_train), (x_test, y_test) = mnist.load_data()

   # reshape matrix to vector
   x_train, x_test = x_train.reshape(x_train.shape[0], -1),
x_test.reshape(x_test.shape[0], -1)

   # scale values to 0-1
   x_train = x_train/255.0
   x_test = x_test/255.0

   train_data, train_labels = [], []
   for i in range(10):
       train_labels += [i]*1000
       temp_data = x_train[np.where(y_train == i)[0]]
       np.random.shuffle(temp_data)
       train_data.extend(temp_data[:1000])

   train_data, train_labels = np.array(train_data), np.array(train_labels)
   idx = np.random.permutation(len(train_data))
   train_data, train_labels = train_data[idx], train_labels[idx]

   idx = np.random.permutation(len(x_test))
   test_data, test_labels = x_test[idx][:1000], y_test[idx][:1000]
   return (train_data, train_labels), (test_data, test_labels)

(x_train, y_train), (x_test, y_test) = get_data()
print(f"Training examples = {x_train.shape}, {y_train.shape}")
print(f"Testing examples = {x_test.shape}, {y_test.shape}")
```

```python
def fit(x_train, y_train):
    models = []
    for i in range(10):
        encode = np.zeros_like(y_train)
        encode[np.where(y_train == i)] = 1
        params = np.linalg.pinv(x_train.T @ x_train) @ x_train.T @ encode
        models.append(params)
    models = np.array(models)
    preds = np.array([np.argmax([x @ model for model in models]) for x in
x_train])
    cm = confusion_matrix(y_train,preds)
    print("Confusion Matrix:")
    print(cm)
    correct = 0
    for i in range(10):
        correct += cm[i][i]
    print(f"Accuracy: {(correct/100)}%")
    return models
models = fit(x_train, y_train)
```

```python
def test(x_test, y_test, models):
    preds = np.array([np.argmax([x @ model for model in models]) for x in
x_test])
    cm = confusion_matrix(y_test,preds)
    print("Confusion Matrix:")
    print(cm)
    correct = 0
    for i in range(10):
        correct += cm[i][i]
    print(f"Accuracy: {(correct/10)}%")
test(x_test, y_test, models)
```