Introduction to ML
Exercise 3

Due Date: December 14th 22:00, 2020

Roni Chernyak, Yosi Shrem, Yael Segal and Joseph Keshet

November 29th, 2021

Guidelines

1. You are not allowed to use external packages other than numpy and scipy.
2. Technical questions about this exercise should be asked at the course' piazza or during the recitation.
3. Personal issues regarding the deadline should be directed to **Roni Chernyak**.
4. In order to submit your solution please submit the following files:
   - (a) `details.txt` - A text file with your full name (in the first line) and ID (in the second line).
   - (b) `ex3.py` - A python 3.6+ file that contains your main function (attach ANY additional files needed for your code to run).
   - (c) `ex_3_report.pdf` - A pdf file in which you describe your model and parameters.
   - (d) `test_y` - your model's predictions on the given test set (see instructions below).

Follow the instructions and submit all files needed for your code to run.

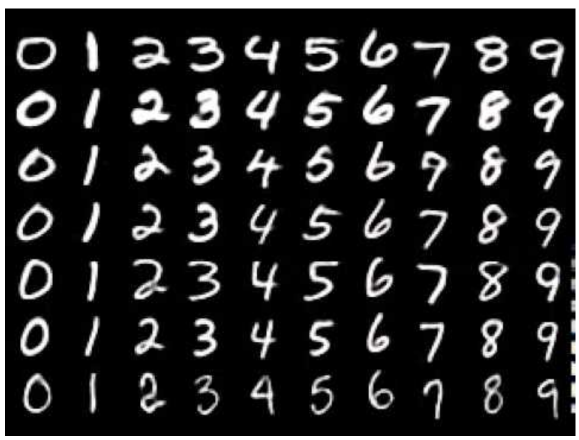
Good Luck!

Ex3

In this exercise you will train your first neural network on a dataset called "MNIST". The dataset contains grayscale images of 10 handwritten digits: 0-9. Your task is to train a classifier that classifies this data.

Data. Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel. This pixel-value is an integer between 0 and 255.

Labels. The labels are the numbers from 0 to 9.



Instructions

1. Your goal is to train a multi-class neural network for the Fashion-MNIST dataset. Your network should have **at least** one hidden layer with the Sigmoid activation function.

2. Your model should minimize the **Negative Log Likelihood** (NLL) loss function as seen in class.(Note: make the adjustments for the Multi-Class).

3. You will receive the data in the form of 3 files: (i) `train_x` will contain the training set examples; (ii) `train_y` will contain the corresponding training set labels; and (iii) `test_x` will contain the test set examples.

4. Your code should get as input three arguments. The first one will be the training examples (`train_x`), the second one is the training labels (`train_y`), and the third one will be the testing examples (`test_x`), where `train_x` and `test_x` will have the same format. For example:

   ```
   $ python ex3.py <train_x> <train_y> <test_x>
   ```

5. You should train and validate your model using files (i)+(ii). Finally, you should output your model's predictions on the examples in `test_x` to a file named `test_y` using the same format as in `train_y` (e.g., row 10 in `test_y` should correspond to the example in row 10 of `test_x`). Your prediction file should contain exactly 5000 rows .

6. You can load the provided data files using `train_x = numpy.loadtxt("train_x")` (repeat this for any data file you wish to load). Suggestion: save a small portion of the dataset and use it for debugging. When you are done, load the entire dataset to train your model.

7. For visualization you can use the following snippet:

   ```
   import matplotlib.pyplot as plt
   plt.imshow(train_x[0].reshape(28,-1),cmap='gray')
   plt.show()
   ```

8. Describe your network's architecture and explain your hyper-parameters choice in a **single** page report called `ex_3_report.pdf`.

9. Submit **ALL** source code files along with your predictions file `test_y`. Make sure to follow the specified format. Your grade will be based on your performance on the test set.

10. **Check the feedback mail to see if there are any errors.** The feed mail will also include a basic sanity check - your accuracy on a random subset containing 100 examples from the test set.

```
Note: to prevent over/under flow, substract max(x) from x as follows.
numerically stable softmax:
def softmax(x):# x is a 1D vec.
    x = x - max(x)
    ...
    ...
    continue with calculation as usual
```

Good Luck!