Introduction to ML
Assignment no. 3

Due: May 17th, 2018 (no extensions)

1 Submission Instructions:

1. **Practical Part**

   (a) You are not allowed to use any machine learning packages or tools (e.g. scikit-learn, PyBrain, PyML, etc.).

   (b) You are allowed to use numpy package

   (c) Use Python 2.7

   (d) In order to submit your solution - please submit the relevant files to the corresponding assignment on the Submit system.

   Your files should include:

   i. `details.txt` - A text file with your full name (in the first line) and ID (in the second line).

   ii. `ex_3.py` - Your code file.

   iii. ex_3_report.pdf - A pdf file in which you describe your model and parameters, and explain why.(1 page at most)

   iv. test.pred - Your prediction for the test file.

   Make sure to name your files correctly.

Good Luck!

2 Practical Part:

In this exercise you will solve a clothing picture classification task using neural networks. We will use the Fashion-MNIST, a well known dataset. both train and test sets, are under the resources section in the piazza.

Content

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, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255.

Labels

The possible labels are:

0. T-shirt/top
1. Trouser
2. Pullover
3. Dress
4. Coat
5. Sandal
6. Shirt
7. Sneaker
8. Bag
9. Ankle boot

Instructions:

Train a multi-class NN with one hidden layer. The loss function is the **negative log likelihood**. Tip: You can use the gradient computations from lecture 6 as reference. To make sure your computations are correct, use the gradient checker.

During the training phase, try multiple hyper-parameters: layer size, learning rate, activation function, batch-size, etc. Report your results in the PDF.

Train and test datasets are under the resources section on piazza. Split the train into train and validation with 80:20 ratio. Use the validation set for hyper-parameters tuning.

Read data:

```
import numpy as np
train_x = np.loadtxt("train_x")
train_y = np.loadtxt("train_y")
test_x = np.loadtxt("test_x")
```

Using your trained classifier, create a `test.pred` file. This file will store your model's predictions for '`test_x`'.
The prediction file should be 5000 lines. Each line is your prediction for the corresponding example in the test file. The grade is based on your performance on the test set.