EE 228 HW#1 - Linear MNIST Classifier

This exercise will focus on training a linear classifier for the MNIST dataset.

Background: MNIST is a standard digit classification dataset. Given a 28×28 image containing a digit from 0 to 9, our goal is deducing which digit the image corresponds to. The dataset has 50,000 training and 10,000 test examples. The data can be downloaded from http://yann.lecun.com/exdb/mnist/.

Formatting the data: Your goal will be building a linear classifier for MNIST. Towards this goal, we need to format the data to obtain a dataset $\mathcal{S} = (x_i, y_i)_{i=1}^{N=50,000}$.

- **Input:** Each input x is a 28×28 matrix. Convert inputs x to vectors of size 784.
- **Output:** Each label y is a digit from 0 to 9. Apply one-hot encoding on y and convert it to a vector y^{oh} of size 10. For instance,

$$y = 5 \quad \text{maps to} \quad y^{oh} = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]$$

Linear classifier: Our goal is learning a matrix $W \in \mathbb{R}^{10\times 784}$ so that given an input x, $f(x) = Wx$ will correctly predict the associated label y. The label $\hat{y}$ output by f is the location of the largest entry of f i.e.

$$\hat{y}_i = \arg\max_{0\le i\le 9} f_i(x)$$

Hence, the test accuracy is given by $\text{acc}(W) = \mathbb{P}(y = \hat{y})$.

Training: You will use quadratic loss and gradient descent for training. The training loss function at ith example (x_i, y_i) is

$$\mathcal{L}_i(W) = \frac{1}{2}\|y_i^{oh} - Wx_i\|_{\ell_2}^2.$$

For training, you will use minibatch stochastic gradient descent (SGD). SGD has three parameters:

- Number of iterations ITR
- a batch size B where $N = 50,000 \ge B \ge 1$.
- a learning rate $\eta > 0$.

SGD algorithm is as follows:

Initialize: $W_0 = 0$

for $1 \le t \le \text{ITR}$:

(i) select B numbers $(r_i)_{i=1}^B$ from $\{1, 2, \ldots, N\}$ uniformly at random (with replacement)

(ii) calculate the gradient $G = \frac{1}{B}\sum_{i=1}^{B} \nabla\mathcal{L}_{r_i}(W)$.

(iii) $W_t = W_{t-1} - \eta G$.

Return $W_{\text{FINAL}} = W_{\text{ITR}}$.

Your tasks are as follows.

1. (2 pts) Write the code for downloading and formatting the data.
2. (5 pts) Write the code for minibatch SGD implementation for your linear MNIST classifier.
3. (7 pts) **The role of batch size:** Run your code with batch sizes $B = 1, 10, 100, 1000$. For each batch size,
   - determine a good choice of learning rate
   - pick ITR sufficiently large to ensure the (approximate) convergence of the training loss
   - Plot the progress of training loss (y-axis) as a function of the iteration counter t (x-axis)
   - Report how long the training takes (in seconds).
   - Plot the progress of the test accuracy (y-axis) as a function of the iteration counter t (x-axis)
4. Comment on the role of batch size.
5. (6 pts) **The role of training dataset size:** Let us reduce the training dataset size. Instead of $N = 50,000$, let us pick a subset $\mathcal{S}'$ of size N' from the original dataset without replacement and uniformly at random. Fix batch size to $B = 100$. Repeat the steps above for $N' \in \{100, 500, 1000, 10000\}$. Comment on the accuracy as a function of dataset size.
6. (Bonus 5 pts) **Simpler Life:** Run the linear MNIST classifier with batchsize $B = 100$ over the full dataset by using PyTorch or Tensorflow. Use same learning rate and initialization $W_0 = 0$. Verify that it is consistent with your handcoded algorithm by comparing your results (the accuracy and training loss plots).