Problem Set 4
ECE 590 Fall 2019

Instructor: Vahid Tarokh
ECE Department, Duke University

Due: 8:59:59 a.m. EST on Oct. 7, 2019

Problem 1: First-order Optimization Methods

In this problem, we want to fit a Logistic Regression model (see Fig. 1) to the MNIST data set using $\mathcal{L}_2$-regularized cross-entropy loss function.

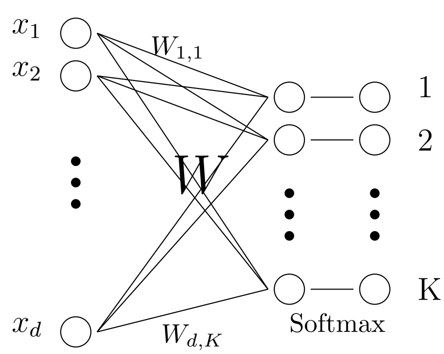

Figure 1: Logistic Regression model

Let K be the number of classes (for MNIST $K = 10$) and let $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}, n = 1, \dots, N$ denote the data set where $\mathbf{y}_n$ is the label of the n-th data point, represented as a K-dimensional vector such that its k-th entry is 1 if the data point belongs to class k and 0 otherwise (i.e., 1-of-K encoding) whereas $\mathbf{x}_n$ is the corresponding d-dimensional feature vector. Then, the cross-entropy loss can be written as:

$$L(\mathcal{D}; \mathbf{W}) = \sum_{n=1}^{N} -\mathbf{y}_n^T \log \text{Softmax}(\mathbf{W}\mathbf{x}_n) + \lambda \|\mathbf{W}\|_F^2, \quad [\text{Softmax}(\mathbf{a})]_i = \frac{\exp{(a_i)}}{\sum_j \exp{(a_j)}}.$$

Write a Python program that fits the model using the following methods:

1. Stochastic gradient descent (SGD)
2. Momentum method with parameter $\beta = .9$
3. Nesterov's accelerated gradient (NAG) method with parameter $\beta = .95$
4. RMSprop with parameters $\beta = .9$, $\gamma = 1$ and $\epsilon = 10^{-8}$
5. Adam with parameters $\beta_1 = .9$, $\beta_2 = .999$ and $\epsilon = 10^{-8}$

with learning rate $\eta = .001$ and batch sizes $\{1, 500, 6 \times 10^4\}$. Report the classification accuracy on the test data set for $\lambda \in \{.1, 1\}$.

Note: You can use the Autograd package from Pytorch to compute the gradient. However, you are **not allowed** to use any built-in optimizers in Python. Instructions on how to download and prepare the data will be given in a Jupyter notebook that will be posted on Sakai.

Problem 2: Newton's Method for Non-linear Optimization

Consider the following non-linear objective function in $\mathbb{R}^2$:

$$F(x_1, x_2) = \exp(x_1 + 3x_2 - 0.1) + \exp(x_1 - 3x_2 - 0.1) + \exp(-x_1 - 0.1).$$

Compute the Hessian $\nabla^2_{(x_1,x_2)} F(x_1, x_2)$.

Write a Python program that implements Newton's method and minimizes $F(x_1, x_2)$ w.r.t. (x_1, x_2). In addition to this, implement Newton's method with a diagonal approximation of the Hessian.

Problem 3: Logistic Regression using Newton's Method

In this problem, we are going to fit a Logistic Regression model on the Breast Cancer data set from problem set 3 (see problem 4) using Newton's method. For this purpose we will use the cross-entropy loss given by:

$$L(\mathcal{D}; \mathbf{w}) = \sum_{n=1}^{N} -y_n \log \sigma(\mathbf{w}^T \mathbf{x}_n) - (1 - y_n) \log(1 - \sigma(\mathbf{w}^T \mathbf{x}_n)), \quad \sigma(a) = \frac{1}{1 - e^{-a}}, a \in \mathbb{R},$$

where $\mathbf{w} = (w_1, \ldots, w_d)$ and $\mathcal{D} = \{\mathbf{x}_n, y_n\}, n = 1, \ldots, N$ where y_n is the label of the n-th data point whereas $\mathbf{x}_n$ is the corresponding feature vector. Compute $\frac{\partial^2 L(\mathcal{D};\mathbf{w})}{\partial w_i \partial w_j}$.

Write a Python program that implements Newton's method to fit the Logistic Regression model on the Breast Cancer data set using the exact expression of the Hessian, computed previously, as well as its diagonal approximation.