TUM EI 70360: Machine Learning and Optimization
Fall 2023
Lecturer: Reinhard Heckel
Teaching Assistant: Tobit Klug

Problem Set 8
Issued: Tuesday, Dec. 5, 2023
Due: Thursday, Dec. 14, 2023

Problem 1 (Equivalence between early stopping and ℓ_2-regularization)**.** Suppose we are given examples $\{(x_i, y_i)\}_{i=1}^n$ that are generated according to $y_i = x_i\theta^* + z_i$ for $i = 1, \dots, n$, where θ^* is the model parameter drawn from a Gaussian prior $\mathcal{N}(0, 1)$ and x_i and z_i are both drawn from a Gaussian distribution with zero mean. Suppose we learn a linear model by minimizing the risk $(1/2)\sum_{i=1}^n (x_i\theta^* - y_i)^2$ with gradient descent. In this setup, is ℓ_2-regularization equivalent to early stopping? Justify your answer.

Problem 2 (Logistic regression with early stopping)**.** In this exercise we explore the effects of regularization with early stopping when training on noisy data. In particular, we consider logistic regression for classifying MNIST digits 5 and 8. However, we are only given noisy observations of the digits, meaning that with probability p, the corresponding label is flipped. The jupyter notebook *logistic_regression_with_early_stopping.ipynb* already contains code to load the data. The training data consists of 500 training samples for each digit, and the test data consists of 500 examples of each digits. Note that 30% of the labels are flipped.

1. Implement logistic regression for this problem. Plot the training and test error as a function of the number of gradient descent steps. You may want to try different stepsizes for gradient descent.

2. Next consider logistic regression with a random feature model. To this end expand each observation $\mathbf{x} \in \mathbb{R}^d$ as follows

$$\mathbf{x}_{rf} = \mathrm{ReLU}(\mathbf{F}\mathbf{x}),$$

   where ReLU is the rectified linear unit function and $\mathbf{F} \in \mathbb{R}^{q\times d}$ is a fixed matrix but with entries drawn randomly from a Gaussian distribution. In case of MNIST digits we have $d = 784$. Thus, choosing $q = 2 \cdot 784$ doubles the number of features for each observation and consequently also the number of trainable parameter of the logistic regression model meaning that we moved to a more overparameterized regime. Add the random feature model to your implementation and again plot the training and test error as a function of the number of gradient descent steps. What do you observe? You may want to normalize the data after applying the random feature model and adjust the stepsize.