CS 461: Machine Learning Principles
Fall 2021

Problem Set #2 (80 Points)

Out: Thursday, September 30

Due: Thursday, October 14, 8:00pm

Instructions

How and what to submit? Please submit your solutions electronically via Canvas. Please submit two files:

1. A PDF file with the written component of your solution including derivations, explanations, etc. You should create this PDF in LaTeX. Please name this document ⟨`firstname-lastname`⟩`-sol2.pdf`.

2. The empirical component of the solution (Python code and the documentation of the experiments you are asked to run, including figures) in a Jupyter notebook file. Rename the notebook ⟨`firstname-lastname`⟩`-sol2.ipynb`.

Late submissions: there will be a penalty of 20 points for any solution submitted within 24 hours past the deadline. No submissions will be accepted past then.

What is the required level of detail? See below.

(a) When asked to **derive** something, please clearly state the assumptions, if any, and strive for balance: justify any non-obvious steps, but try to avoid superfluous explanations.

(b) When asked to **plot** something, please include in the `ipynb` file the figure as well as the code used to plot it. If multiple entities appear on a plot, make sure that they are clearly distinguishable (by color or style of lines and markers) and references in a legend or in a caption.

(c) When asked to **provide a brief explanation or description**, try to make your answers concise, but do not omit anything you believe is important. If there is a mathematical answer, provide it precisely (and accompany by succinct wording, if appropriate).

(d) When **submitting code (in Jupyter notebook)**, please make sure it's reasonably well-documented, runs, and produces all the requested results. If **discussion** is required/warranted, you should include it directly in the notebook (using the markdown cell).

Collaboration policy: collaboration is allowed and encouraged, as long as you (1) write your own solution entirely on your own, and (2) specify names of student(s) you collaborated with in your writeup.

1 Classification

The written part of the assignment will focus on linear classifiers. In the lecture, we discussed binary classification in depth where there are only two possible labels: $y \in \{1, 2\}$. The **logistic regression model** has a learnable parameter vector $w \in \mathbb{R}^d$ and calculates the conditional probability of label 1 given an d-dimensional input $x \in \mathbb{R}^d$ by

$$p_w(1|x) = \sigma(w^\top x) = \frac{1}{1 + \exp(-w^\top x)}$$

where $\sigma : \mathbb{R} \to (0, 1)$ is the **sigmoid** function that squashes any real value to a probability. This also yields the conditional probability of label 2 as $p_w(2|x) = 1 - p_w(1|x)$. We saw how this can be derived as modeling the *log-odds* as a linear function $\log \frac{p_w(1|x)}{p_w(2|x)} = w^\top x$. We now focus on multiclass classification where there are $L \geq 2$ labels: $y \in \{1 \dots L\}$. The **softmax model** has a learnable parameter matrix $W \in \mathbb{R}^{d \times L}$, equivalently L parameter vectors as columns of $W = (w_1 \dots w_L)$, and calculates the conditional probability of label y given an d-dimensional input $x \in \mathbb{R}^d$ by

$$p_W(y|x) = \text{softmax}_y((w_1^\top x, \dots, w_L^\top x)) = \frac{\exp(w_y^\top x)}{\sum_{y'=1}^{L} \exp(w_{y'}^\top x)}$$

where $\text{softmax} : \mathbb{R}^L \to (0, 1)^L$ is the **softmax** function that squashes any real-valued vector into a distribution over its dimensions.

Problem 1 **[10 points]**
Show that the softmax model implies modeling the log-odds between any two labels $y, y' \in \{1 \dots L\}$ by a linear function.

End of problem 1

Problem 2 **[5 points]**
Suppose $L = 2$. Given any softmax model with parameter vectors $w_1, w_2 \in \mathbb{R}^d$, show an equivalent logistic regression model with parameter vector $w \in \mathbb{R}^d$. That is, express w as a function of w_1 and w_2 so that the conditional label probabilities are the same under each model.

End of problem 2

Problem 3 **[10 points]**
More generally, show that the softmax model is "overparameterized" as follows. Given any softmax model with L parameter vectors $w_1 \dots w_L \in \mathbb{R}^d$, show an equivalent softmax model with $L - 1$ nonzero parameter vectors $v_1 \dots v_{L-1} \in \mathbb{R}^d$. Explain how this implies that we need only $L - 1$, not L, learnable parameter vectors in a softmax model with L labels.

End of problem 3

Problem 4 **[15 points]**
(This problem will be directly useful in the coding part.) Let $(x_1, y_1) \ldots (x_N, y_N) \in \mathbb{R}^d \times \{1 \ldots L\}$ denote a batch of N labeled examples for classification. The cross-entropy loss on this batch with l_2 regularization for the softmax model $W \in \mathbb{R}^{d \times L}$ is

$$\widehat{J}(W) = -\frac{1}{N} \sum_{i=1}^{N} \log p_W(y_i|x_i) + \lambda \sum_{j=1}^{d} \sum_{l=1}^{L} W_{j,l}^2$$

where $\lambda \geq 0$ is a constant hyperparameter for controlling the strength of regularization. We will omit the bias parameter WLG assuming the first input dimension has value 1. Calculate the gradient $\nabla \widehat{J}(W) \in \mathbb{R}^{d \times L}$ in *matrix form*. It should be a function of

- Data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ where the i-th row corresponds to $x_i \in \mathbb{R}^d$
- Model probability matrix $P \in \mathbb{R}^{N \times L}$ where $P_{i,l} = p_W(l|x_i)$
- Gold label matrix $G \in \mathbb{R}^{N \times L}$ where $G_{i,l} = 1$ if $y_i = l$ and $G_{i,l} = 0$ otherwise.
- Parameter matrix $W \in \mathbb{R}^{d \times L}$

only using matrix operations (i.e., do not enumerate elements of any matrix).

End of problem 4

Advice: First calculate the gradient of $\widehat{J}$ assuming $N = 1$ with respect to an individual parameter vector $w_l \in \mathbb{R}^d$, then put it together in matrix form.

2 Digit Classification (MNIST)

We now apply the softmax model to the problem of classifying handwritten digits.

Dataset We will work with the MNIST dataset. This is a dataset of handwritten digits which has served as a machine learning benchmark for many years. Each example is a 28-by-28 pixel grayscale image of a handwritten digit; we will be working with a vectorized representation of the images, converted to a 784-dimensional integer vector with values between 0 (white) and 255 (black). The dataset provided to you consists of three parts:

- **Training** set of 20000 examples
- **Validation** set of 10000 examples
- **Test** set of 10000 examples (no labels are provided)

Each set is divided equally among 10 classes. In addition to the normal training set of 20000 examples, we include a **small training** set of only 30 examples (3 per class), to investigate the effect of dataset size on training a classifier in this domain.

Problem 5 **[10 points]**
Implement the negative log likelihood calculation inside the `forward` function of the model class.

End of problem 5

Problem 6 **[20 points]**
Implement the gradient calculation inside the `accumulate_gradients` function of the model class.

End of problem 6

Problem 7 **[10 points]**
Answer discussion questions directly in the markdown cells (look for "Question" headings). The topics are

- Plot and analysis: Confusion matrix, model weight visualization
- Comparing the models across the two data regimes (large and small)
- Showing the performance of the best model: Training and validation accuracy, test accuracy (from Kaggle submission, see below)

End of problem 7

2.1 Kaggle Submission

We have set up a Kaggle competition to which you will be submitting your final predictions on the test set (link). First, you will have to create a Kaggle account (with your email: `scarletmail.rutgers.edu`). Once you have access to the competition page, read through all three information pages carefully (Description, Evaluation, and Rules) and accept the rules. You will now be ready to make submissions. The notebook creates a file `cs461hw2_mynetid.csv` in your Google Drive where `mynetid` is your NetID. Once you have accepted the Kaggle rules, there will be an option to "Make a submission", where you can upload this CSV file. To make sure you do not overfit this held-out set, we have limited your submissions to two per day, so start early and you will get more chances if you make mistakes. Your up to date score will appear on the public leaderboard once you submit.