Assignment 3

September 2021

Instructions

- This assignment should be completed individually.
- Do not look at solutions to this assignment or related ones on the Internet.
- The files related to the assignment are present in `lab3-rollno.zip` folder. Extract it and upload it on moodle in the same .zip format after completion and after replacing the string "rollno" with your actual roll number. For example, if you roll number is 00405036, then single zip folder that you will upload will be named "lab3-00405036.zip". Also collate all the CS337 based theory solutions into ONE pdf file named `answers.pdf`. Include `answers.pdf` inside the zip folder mentioned above and submit the zip folder.
- Answers to all subjective questions need to be placed in single pdf `answers.pdf` including all plots and figures and uploaded.
- Only add/modify code between `TODO` and `END TODO` unless specified otherwise. You must not import any additional libraries.
- Python files to submit - `assignment_3.ipynb`
- This Assignment carries a total of 16.5 marks for CS337 Theory and 11.5 marks for CS335 Lab

1 Logistic Regression

1.1 CS 337: Logistic Regression

Consider the following formulation for extending the logistic regression to a multi-class classification setup:

$$P(Y = k|\mathbf{w}_k, \phi(\mathbf{x})) = \frac{e^{\mathbf{w}_k^T \phi(\mathbf{x})}}{\sum_{k=1}^{K} e^{\mathbf{w}_k^T \phi(\mathbf{x})}}$$

Here, each $\mathbf{w}_k$ is a class specific vector of dimension equal to the number of features in $\phi(x)$. This expression is called softmax. Note that this expression is slightly different from the multi-class

logistic mentioned in Lecture 9 Each of the weight vectors $\mathbf{w}_k$ are computed by optimizing the categorical cross-entropy loss function.

$$E(\mathbf{W}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_k^{(i)} \log(P(Y = k|\mathbf{w}_k, \phi(\mathbf{x}^{(i)})))$$

Here $y^{(i)}$ is a K length one-hot vector representing the label of the i^{th} example, and $\mathbf{W}$ is a matrix having $\mathbf{w_k}$ as its columns. Note that $\mathbf{W}$ is a matrix of dimensions (num_features x num_classes).

Show that cross entropy used to train a binary logistic regression (Eqn (3) of Lecture 9) is a special case of this. (1 mark)

1.2 Logistic Regression's Decision surface

For this question, let us assume a 2 class dataset with features $\mathbf{x} \in R^d$ and labels $y \in -1, +1$. The decision rule obtained from a 2-class Logistic regression model with decision threshold $\theta \in (0, 1)$ is given by

$$P(y = +1|\mathbf{x}) = \frac{1}{1 + exp(-\mathbf{w}^T\mathbf{x})} \geq \theta$$

For simplicity let us assume that $\theta = 0.5$. We know that the Sigmoid function, ($f(s) = \frac{1}{1+e^{-s}}$ where $s = \mathbf{w}^T\mathbf{x}$), as shown in the equation above is non-linear in $\mathbf{w}$. Despite that, we call Logistic Regression a linear model. Argue why that is the case. (1 mark)

1.3 CS 337: Multi Class Logistic Regression

Note

In this question, we will derive the gradients for Multi-class logistic regression model building gradually from the concepts we have learnt so far in the class.
Caution - While the solution to this question is easy, it may involve a lot of notational clutter. We request to stick to the notations introduced here and mention all the steps neatly, in details.
Throughout this question we will use

- i to index training examples
- j to index features of a given example
- k to index classes

For the theory part, let us assume that there is no bias parameter.

Let us assume a 3 class classification problem, *i.e.*, $K = 3$ and $Y \in \{-1, 0, +1\}$. The training data X contains text samples. You can think of (x, y) pair from training data to be of the form $(text, sentiment)$, with an example being: (The food in the restaurant tastes good, $+1$). Similarly there may be other text examples for the negative class (-1) as well as for the neutral classes $(+1)$. Each text x is a sequence of words sampled from a dictionary containing 1000 words. *i.e.*, the vocabulary size is 1000

Given the training data X, Y, we derive several binary features from it using a set of feature functions $\{\phi\}$. For the scope of this question assume that each ϕ_j is a binary Indicator function indexed by words in the dictionary in a lexicographically sorted order. Thus $\phi(X)$ is a 1000 dimensional binary vector. $\phi_j(x) = 1$ if the j^{th} word is present in x (irrespective of the number of times the word appears) and 0 otherwise. These are popularly referred to as the Bag of Words features (without count values).

Qn (a) Specify what the feature vector ϕ would look like when

- $x =$"The food in the restaurant tastes good"

(0.5 marks)

Qn (b) Mention two limitations or drawbacks of this featurization technique, apart from the point that this does not keep track of the count of words (0.5 marks)

Qn (c) Suppose we learn the Logistic Regression model parameters using the standard MLE approach. In that case, how many parameters are to be learned (0.5 marks)

Qn (d) Assume that we have the optimal parameters w_0^*, w_1^*, w_2^* available with us. One simple way perhaps could have been to use *sum normalization* to obtain the posterior probabilities as shown below:

$$P(y = k|x) = \frac{w_k^{*T} x}{\sum_{k=0}^{2} w_k^{*T} x}$$

Identify one serious flaw with such as sum normalization approach (0.5 marks)

Qn (e) Let us now solve a numerical problem.
Assume that $[w_0^{*T} x, w_1^{*T} x, w_2^{*T} x] = [0.1, 0.5, 2]$. Obtain the posterior values using (i) Sum normalization (ii) softmax normalization. Compare and comment on the values that you obtain. (0.5 + 0.5 marks)

Qn (f) Given n training data examples $D = \{(x_i, y_i)\}_{i=1}^n$, write the expression for the data likelihood $P(D|W)$. Capital W denotes the set of all the parameters. (2 marks)

Qn (g) Next, we will maximize the log data likelihood *w.r.t.* the parameters of the model. Write the expression for log data likelihood as the difference of two terms, *i.e.*, $\log P(D|W) = Numer - Denom$ where $Numer = Numerator$ and $Denom = denominator$ (1 mark)

Qn (h) Next, let us compute the gradient. Compute $\frac{\partial Numer}{\partial w_j}$. Here, w_j is the weight of the j^{th} feature ϕ_j. Simplify as much as you can. (2 marks)

Qn (i) This is a tricky part. Please be careful with the notation write your answer thoughtfully. Compute $\frac{\partial Denom}{\partial w_j}$. w_j is the weight of the j^{th} feature ϕ_j. Simplify as much as you can. (3 marks)

Qn (j) Consolidate the expressions from Qn (i) and Qn (j) and thus write the expression for $\frac{\partial \log P(D|W)}{w_j}$ (0.5 marks)

Qn (k) We may now use the First Order Optimality condition and equate the gradient in Qn (k) to 0 at optimal W^*, *i.e.*,

$$x^* = argmax_x f(x) \implies \frac{\partial f}{\partial x}|_{x=x^*} = 0$$

Apply the first order optimality condition and obtain an expression that has some nice interpretation. If all your steps are correct, you should get what are called as **Balance Equations**. (2 marks)

Qn (l) Analyze the expression in Qn (k). Carefully analyse the expression for a while, and explain in english what it conveys (1 mark)

1.4 CS 335: Lab Problems

(a) In this problem, we will try to predict the digit shown in the image. The dataset `train_X.npy` contains features for images of size 28x28 as numpy array, `train_y.npy` contains digits for corresponding images, for simplicity we will work with only for digits i.e. 1, 4, 7 and 9. We will implement logistic regression for this multi-class classification problem by making `one_vs_all` binary logistic regression classifiers for each of the 4 classes. Perform the following tasks.

(i) Complete the functions `normalize()` and `scaling()` in `Assignment_3.ipynb` to perform Gaussian Normalization and MinMax Scaling on input image features provided in `train_X.npy`. Try using normalization and scaling inside the function `split_data()`. (0.5 marks)

(ii) Complete the function `get_data_for_class()` in `Assignment_3.ipynb` which takes class number as input and modifies the given dataset according that class. (1 marks)

(iii) Complete the function `sample_training_points()` in `Assignment_3.ipynb` which samples a random batch of size `sample_size` from given dataset. (0.5 marks)

(iv) Complete the functions `sigmoid()`, `cross_entropy_loss()` and `grad()` in `Assignment_3.ipynb` which computes logistic sigmoid, binary cross entropy and gradient of parameters respectively. (3 marks)

(v) Complete the function `logistic_regression` in `Assignment_3.ipynb` which trains a `one_vs_all` binary logisitc regression classifier for a particular class and return its parameters. (2 marks)

(vi) Complete the function `prediction()` which computes the probablity of a class on the basis of parameters of `one_vs_all` classifier for that class. (1 marks)

(vii) Complete the function `accuracy()` which calculates the accuracy for each classifier.(0.5 marks)

Make sure you write an efficient vectorized implementation for each task. You are allowed to change the learning rate `lr`, maximum iterations `epochs` and sample size `smaple_size` in function `train_multi_class()` in `Assignment_3.ipynb`.

(b) Report the validation accuracy you obtained. Now, consider a model M which predicts digit 1 for any image. What accuracy does this model achieve on the validation set? Do you think accuracy is a good evaluation metric here? Briefly justify your answer. (1 marks)

(c) Consider a different evaluation metric F_1 score, defined as the harmonic mean of precision and recall. Precision is defined as the fraction of positive outputs which are actually positive. Recall is defined as the fraction of actually positive samples predicted as positive. Formally, let's denote TP as true positives, FP as false positives and FN as false negatives. Then recall, precision and F_1 score are defined as follows:

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{F}_1 = \frac{2 * \text{recall} * \text{precision}}{\text{recall} + \text{precision}}$$

Complete the function `calculate_metrics()`. Report the `precision, recall` and F_1 `score` you obtain on the validation set. Also, report the `precision, recall` and F_1 `score` achieved by the model M described in the previous part. Do you think `precision, recall` and F_1 `score` are good evaluation metrics for this task? Briefly justify your answer. (2 marks)