CSCI 374: Homework 2

Due: Oct 11, 2022 at 11:59pm EST

Name: Kelly McCarthy

Instructions: This homework requires answering some open-ended questions, short proofs, and programming. This is an individual assignment, not group work. Though you may discuss the problems with your classmates, you must solve the problems and write the solutions independently. As stated in the syllabus, copying code from a classmate or the internet (even with minor changes) constitutes plagiarism. You are required to submit your answers in pdf form (use LaTeX) in a file called `hw2.pdf` to Gradescope under "HW2". Code should also be submitted to Gradescope in a file called `hw2.py` under "HW2 Programming". A LaTeX template for the pdf submission and a code skeleton for the code submission are available on Piazza. Please do **not** modify the base code in the coding skeleton; simply augment it with your solution. Late submissions do not receive full credit, except in extenuating circumstances such as medical or family emergency. Submissions submitted 0-24 hours late are eligible for only 90%, 24-48 hours late for 80%, 48-72 hours late for 70%, and later than 72 hours for 0% of the total credit for this assignment. Late days may be used (if available) to avoid these penalties.

The total assignment is worth 85 points.

Problem 1 (20 points)

In HW1 we proved that gradient descent is indeed a descent algorithm (each step produces a lower loss) for any twice differentiable and K-Lipschitz continuous loss function $L(\theta)$, as long as we choose a learning rate $\alpha \leq 1/K$. We will now use this result to prove a stronger form of convergence: convergence to a global optimum when the loss function $L(\theta)$ is also convex (which is often the case for linear and logistic regression using the loss functions we have discussed in class.)

A function $L(\theta)$ is said to be convex if and only if for any two $\theta^{(j)}$ and $\theta^{(k)}$

$$L(\theta^{(j)}) \leq L(\theta^{(k)}) + \nabla L(\theta^{(j)})^T(\theta^{(j)} - \theta^{(k)}).$$

For the following questions, also recall that $||A||^2 = A^T A$.

(i) Let θ^* denote the global minimum of $L(\theta)$. On HW1, we showed that for $\alpha \leq 1/K$

$$L(\theta^{(i+1)}) \leq L(\theta^{(i)}) - \frac{\alpha}{2}||\nabla L(\theta^{(i)})||^2$$

Use the definition of convexity in the above expression, plugging in $\theta^{(i)}$ and θ^* for $\theta^{(j)}$ and $\theta^{(k)}$ respectively, to prove the following inequality (10 points):

$$L(\theta^{(i+1)}) - L(\theta^*) \leq \frac{1}{2\alpha}\big(||\theta^{(i)} - \theta^*||^2 - ||\theta^{(i+1)} - \theta^*||^2\big).$$

Hint 1: For scalars a, b we have $(a-b)^2 = a^2 - 2ab + b^2$. Similarly for vectors A, B of equal length, we have $||A-B||^2 = ||A||^2 - 2A^TB + ||B||^2 = ||A||^2 - 2B^TA + ||B||^2$.
Hint 2: Keep in mind the relation between $\theta^{(i)}$ and $\theta^{(i+1)}$ based on gradient descent.

$L(\theta^{(i+1)}) \leq L(\theta^{(i)}) - \frac{\alpha}{2}||\nabla L(\theta^{(i)})||^2$
$L(\theta^{(i+1)}) \leq L(\theta^{(*)}) + \nabla L(\theta^{(i)})^T(\theta^{(i)} - \theta^{(*)}) - \frac{\alpha}{2}||\nabla L(\theta^{(i)})||^2$

Let a = $\alpha \nabla L(\theta^{(i)})$

$b = (\theta^{(i)} - \theta^{(*)})$

$\leq -(\frac{\alpha}{2})||\nabla L(\theta^{(i)})||^2 + \nabla L(\theta^{(i)})^T(\theta^{(i)} - \theta^{(*)}) * 2\alpha/2\alpha$

$\leq \frac{1}{2\alpha}(-(\frac{2\alpha^2}{2})||\nabla L(\theta^{(i)})||^2 + 2\alpha \nabla L(\theta^{(i)})^T(\theta^{(i)} - \theta^{(*)})$

$\leq \frac{1}{2\alpha}(-\alpha^2||\nabla L(\theta^{(i)})||^2 + 2\alpha \nabla L(\theta^{(i)})^T(\theta^{(i)} - \theta^{(*)}))$

$\leq \frac{1}{2\alpha}(-||\alpha \nabla L(\theta^{(i)})||^2 + 2\alpha \nabla L(\theta^{(i)})^T(\theta^{(i)} - \theta^{(*)}))$

$\leq -\frac{1}{2\alpha}((-||a||^2 + 2a^Tb) + ||(\theta^{(i)} - \theta^{(*)})||^2 - ||(\theta^{(i)} - \theta^{(*)}))||^2$

$\leq -\frac{1}{2\alpha}((||a||^2 + 2a^Tb - ||b||^2) + ||b||^2)$

$\leq -\frac{1}{2\alpha}((||B - A||^2 - ||B||^2))$

$\leq -\frac{1}{2\alpha}((||(\theta^{(i)} - \theta^{(*)}) - (\alpha \nabla L(\theta^{(i)}))||^2 - ||(\theta^{(i)} - \theta^{(*)})||^2))$

Note: manipulate update rule and plug in for:$(\alpha \nabla L(\theta^{(i)}))$
$\alpha \nabla L(\theta^{(i)}) = (\theta^{(i)} - \theta^{(i+1)})$

$\leq -\frac{1}{2\alpha}((||(\theta^{(i)} - \theta^{(*)}) - (\theta^{(i)} - \theta^{(i+1)})||^2 - ||(\theta^{(i)} - \theta^{(*)})||^2))$

Distribute negative sign and eliminate terms

$\leq \frac{1}{2\alpha}((||(\theta^{(i+1)} - \theta^{(*)})||^2 - ||(\theta^{(i)} - \theta^{(*)})||^2))$

(ii) Suppose we start gradient descent at some initial point $\theta^{(0)}$ and continue for N steps, reaching the point

$\theta^{(N)}$. We now prove a bound on the difference between the loss at $\theta^{(N)}$ and the optimal point θ^*. Since the loss function decreases at every step, we know that the following inequality holds:

$$L(\theta^{(N)}) - L(\theta^*) \leq \frac{1}{N} \sum_{i=0}^{N-1} \big(L(\theta^{(i+1)}) - L(\theta^*)\big).$$

Intuitively, the above inequality just says that the loss after the Nth iteration is less than or equal to the loss at previous iterations, and thus the mean across all iterations. Prove that this and (i) imply (6 points):

$$L(\theta^{(N)}) - L(\theta^*) \leq \frac{||\theta^{(0)} - \theta^*||^2}{2\alpha N}.$$

(1) $\mathrm{L}(\theta^{(N)}) - L(\theta^*) \leq \frac{1}{N} \sum_{i=0}^{N-1} \big(L(\theta^{(i+1)}) - L(\theta^*)\big)$

(2) From part (i): $\mathrm{L}(\theta^{(i+1)}) - L(\theta^*) \leq \frac{1}{2\alpha}\big(||\theta^{(i)} - \theta^*||^2 - ||\theta^{(i+1)} - \theta^*||^2\big)$.

(3) $\mathrm{L}(\theta^{(N)}) - L(\theta^*) \leq \frac{1}{N} \sum_{i=0}^{N-1} \big(L(\theta^{(N)}) - L(\theta^*)\big)$

(4) $\leq \frac{1}{N} \sum_{i=0}^{N-1} \frac{1}{2\alpha}\big(||\theta^{(N)} - \theta^{(*)}||^2 - ||\theta^{(N+1)} - \theta^{(*)}||^2\big)\big)$

(5) $\leq \frac{1}{2\alpha N} \sum_{i=0}^{N-1} \big(||\theta^{(0)} - \theta^{(*)}||^2 - ||\theta^{(N)} - \theta^{(*)}||^2\big)\big)$

(6) $\leq \frac{||\theta^{(0)} - \theta^{(*)}||^2}{2\alpha N}$

(7) $\mathrm{L}(\theta^{(N)}) - L(\theta^*) \leq \frac{||\theta^{(0)} - \theta^{(*)}||^2}{2\alpha N}$

Note for step (4):
This is a telescoping sum, subsequent terms cancel and we are left with only the initial and final.

Notes for step (5):

For $\frac{1}{2\alpha N}$,we can combine these terms ($\frac{1}{N}$ and $\frac{1}{2\alpha}$) because we are multiplying by a constant.

$||\theta^{(N)} - \theta^{(*)}||^2$
This expression evaluates to some positive constant so we can disregard it in the proceeding step

(iii) Briefly comment on how the final inequality in (ii) establishes proof of the convergence of gradient descent to the optimal solution after enough iterations. (4 points)

From part (i), we know that a function $L(\theta)$ is convex if and only if for any two θ^j and θ^k, the following holds true:
$L(\theta^{(j)}) \leq L(\theta^{(k)}) + \nabla L(\theta^{(j)})^T(\theta^{(j)} - \theta^{(k)})$
We were able to plug in the above equation where j and k were replaced with θ^i and θ^*, where θ^* is some global minimum. If a global minimum exists for a function, we know this to be a point of convergence.

From part (ii), we learn that the loss after the Nth iteration is less than or equal to the loss at previous iterations. So, the final inequality we proved for part (ii) states that the loss after the Nth iteration is less than or equal to the difference squared of θ^0 (the initial point) and θ^* (the global minimum). Since the greatest distance that exists will be between the initial point and the global minimum, if this inequality states that the loss after the Nth iteration is less than or equal to the value of this squared difference, then the gradient descent has converged after enough iterations.

https://www.overleaf.com/project/6346041148e9b864a47a4c82

Problem 2 (15 points, 5 points per question)

The concept of *priors* is quite popular in machine learning and statistics, particularly in Bayesian thinking. Here, we will prove that L2 regularization is equivalent to encoding a prior belief that the most likely value for each parameter θ_i is 0, and the probability of values other than 0 falls off according to a normal distribution with mean 0 and variance determined by the regularization hyperparameter λ. Similarly, we will prove that L1 regularization is equivalent to imposing a prior belief that each parameter follows a Laplace distribution with mean 0 and scale λ. Take a moment to familiarize yourself with the shape and basic definitions of the Gaussian and Laplace distributions from Wikipedia.

We have previously noted how maximizing the likelihood function $p(Y \mid X, \theta)$ is a reasonable objective. We now see how maximizing the *posterior probability* $p(\theta \mid Y, X)$ leads to regularized models. Applying Bayes rule we have,

$$p(\theta \mid Y, X) = \frac{p(\theta, Y \mid X)}{p(Y \mid X)} = \frac{p(Y \mid X, \theta) \times p(\theta)}{p(Y \mid X)}$$

The second equality follows from applying the chain rule of probability to the numerator and under the assumption that $\theta \perp\!\!\!\perp X$. Since the denominator does not depend on θ, finding a set of parameters that

maximizes $p(\theta \mid Y, X)$ is equivalent to finding a set that maximizes just $p(Y \mid X, \theta) \times p(\theta)$, i.e., the likelihood times the prior.

Hint: When factorizing the joint distribution $p(\theta) \equiv p(\theta_1, \ldots, \theta_d)$ below, you should make use of the fact that $\theta_i \perp\!\!\!\perp \theta_j$ for all distinct i, j pairs, i.e., the parameters are mutually independent.

(i) Under the mean zero Gaussian prior for L2 regularization, we have for each parameter θ_j

$$p(\theta_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\theta_j^2}{\sigma^2}}$$

Prove that

$$\arg\max_\theta \prod_{i=1}^{n} p(y_i \mid x_i, \theta) \times p(\theta) \;=\; \arg\min_\theta - \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + \lambda \sum_{j=1}^{d} \theta_j^2$$

where the first term is the negative log likelihood and the second term is L2 regularization with $\lambda = 1/\sigma^2$.

Proof.

$\arg\max_\theta \prod_{i=1}^{n} p(y_i \mid x_i, \theta) \times p(\theta)$

$= \log(\arg\max_\theta \prod_{i=1}^{n} p(y_i \mid x_i, \theta) + log(p(\theta))$

$= \arg\max_\theta + \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + log(\prod_{i=1}^{n} p(\theta_j)) + log(\prod_{j=1}^{d} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\theta_j^2}{\sigma^2}})$

$= \arg\max_\theta + \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + d \times log(\frac{1}{\sigma\sqrt{2\pi}}) + \frac{1}{\sigma^2} \sum_{j=1}^{d} (-\theta_j)^2$

$= \arg\min_\theta - \sum_{i=1}^{n} log \times p(y_i \mid x_i, \theta) + \lambda \sum_{j=1}^{d} \theta_j^2$

□

(ii) Under the mean zero Laplace prior for L1 regularization, we have for each parameter θ_j

$$p(\theta_j) = \frac{1}{2b} e^{-\frac{|\theta_j|}{b}}$$

Prove that

$$\arg\max_\theta \prod_{i=1}^{n} p(y_i \mid x_i, \theta) \times p(\theta) \;=\; \arg\min_\theta - \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + \lambda \sum_{j=1}^{d} |\theta_j|$$

where the first term is the negative log likelihood and the second term is L1 regularization with $\lambda = 1/b$.

$\arg\max_\theta \prod_{i=1}^{n} p(y_i \mid x_i, \theta) \times p(\theta)$

$= log(\arg\max_\theta \prod_{i=1}^{n} p(y_i \mid x_i, \theta) + log(p(\theta))$

$= \arg\max_\theta \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + log(\prod_{j=1}^{d}(p(\theta_j))$

$= \arg\max_\theta \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + log(\prod_{j=1}^{d} \frac{1}{2b} e^{-\frac{|\theta_j|}{b}} + d \times log(\frac{1}{2b}) + \frac{1}{b} \sum_{j=1}^{d} -|\theta_j|)$

$= -(\arg\max_\theta \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + \lambda \sum_{j=1}^{d} -|\theta_j|)$

$= \arg\min_\theta - \sum_{i=1}^{n} \log p(y_i \mid x_i, \theta) + \lambda \sum_{j=1}^{d} |\theta_j|$

(iii) Based on examining the shape of the Laplace and Gaussian distributions, provide an intuitive explanation of why L1 regularization enforces stronger sparsity (values are pushed to zero quicker) than L2.

L1 regularization = sparse solution = Laplace
L2 regularization = non-sparse solution = Gaussian (normal)

With L1 regularization, since values are pushed to zero quicker, we know this more closely aligns with a Laplacian distribution since the graph shows a distribution of data points in a more confined space, whereas the Guassian (Normal) distribution has a broader range of data points, and thus encourages a non-sparse solution associated with L2 regularization.

Problem 3 (15 points, 3 points per question)

Diagnose the issues in the following scenarios as arising primarily due to (a) underfitting, (b) overfitting, violations of the iid assumption due to (c) lack of independence, or (d) distribution shift (data are not identically distributed.) Provide a brief justification of your answers in 1-2 sentences. More than one answer is possible. The goal is to develop critical thinking on a range of possible issues, and assumptions in the model to investigate.

(i) A regression model with R^2 values of 0.05 and 0.02 during training and testing respectively is deemed unfit for assisting with predicting the severity of forest fires.

(ii) A model used to predict slumps in the British economy fails to predict a downturn caused by events in Ukraine and Russia.

(iii) Google developed a machine learning tool, known as Google Flu Trends, for predicting the number of flu cases in a certain period of time based on popular searches in the same period. The algorithm fit the historical data almost perfectly, but upon launch, it consistently produced erroneous results – at the peak of the 2013 flu season it was off by about 140%.

(iv) A machine learning model built for predicting adverse cardiac events in the ICU using data from the University of Wisconsin hospital system fails to generalize to hospital systems in other parts of the world.

(v) An email spam filter starts marking all emails containing a brand new emoji as suspicious.

(i) Underfitting. When we obtain poor R2 / accuracy on our training data and testing data, the model does not fit well to the data it has been given.

(ii) Violations of the iid (independently and identically distributed) assumption due to distribution shift (data are not identically distributed). One of the iid conditions is that you are "drawing from the same bag," aka you are drawing from the same distribution. Each draw from the distribution must also be independent of the other draws. While the British economy likely has historical data of past wartime conflicts of its own country to train on, it may have limited information about historical data for other countries, causing us to try to make a prediction from one distribution about another distribution. The draws are also not independent.

(iii) Overfitting. We have a good (almost perfect) R2 value on our training data but poor performance on the test set.

(iv) Overfitting. The model fails to generalize to the test set.

(v) Underfitting. The model does poorly on the training set and has poor generalization error to new data.

Problem 4 (13 points)

The following are conceptual questions related to regularization.

(i) Discuss the relative trade offs between L1 and L2 regularization. (4 points)

(ii) Describe one method of picking a suitable value for the regularization hyperparameter λ. (3 points)

(iii) Does there always exist a setting of λ such that we are guaranteed to obtain better results on unseen data? (2 points)

(iv) Discuss why learning rate α does not require the same amount of tuning as λ. (4 points)

(i) L1 regularization favors feature sparsity by pushing parameters that are associated with unimportant features to zero quicker than L2 regularization. L1 is preferred when we are prioritizing feature selection and interpretability of the model. L1 however, is more difficult to work with than L2 because L1 is non-differentiable whereas L2 is differentiable.

Regularization is a common technique used in machine learning to bias a model towards simpler solutions. The core idea behind regularization is to add a penalty related to the size of the parameters.
L1 regularization: We penalize the sum of the absolute values of the parameters
L2 regularization: We penalize the sum of the squares of all parameters

(ii) One method is to split your data into training, validation, and testing sets. When we fit the model on the training set, we will use different values of the hyperparameters and then select the value which maximizes performance on the validation set. Once the model is tried with the test set, we will have a better understanding of its true ability to generalize. This is often referred to as the train-validate-test pipeline.

(iii) No, not necessarily. Since lambda is so sensitive, it is possible that for example, setting it to 0 results in the same overfit curve that existed prior to regularization. When we implement a small amount of regularization, we may prevent overfitting and if we increase the strength again just slightly, it may go back to underfitting. While it is possible to obtain better results on unseen data, it is difficult to pinpoint an exact lambda value that is suitable so it may not always be guaranteed.

(iv) For the learning rate alpha, once theta at i approaches smaller values, the learning rate also gets pushed towards 0 at a slower rate for L2 regularization. For L1, the learning rate updates at either -1 or +1. In this way, the learning rate is less sensitive to alterations. With lambda however, we know that features that do not hold much predictive value for the outcome (y) will have parameters that are pushed closer to zero much more quickly. In this way, we must strategically pick a good lambda value in order to disregard unimportant features and highlight predictive ones.

Problem 5 (22 points)

In this problem you will implement and apply logistic regression with the option of applying L2 regularization based on the code skeleton provided in `hw2.py` and helper functions provided

in `data_loader.py`. There are two datasets provided to help test your implementation: an artificially generated dataset in `simulated_data.csv` with features X_1, X_2, X_3 and outcome Y, and a breast cancer dataset from the University of Wisconsin[1]. The features in the breast cancer data include information about the tumor cells derived from imaging – mean radius, size etc. The outcome of interest is a classification of whether the tumor is malignant (dangerous) or benign.

The helper functions in data loader are meant to assist with your implementation and analysis.

(i) Implement logistic regression and its associated methods using gradient descent using the mean negative loglikelihood function as the loss. Hint: as you iterate on your implementation, you may want to comment out the code corresponding to analyzing the breast cancer data. (10 points)

Honor code note: You can copy some code from your own version of HW1, but not others'.

(ii) Implement an option for L2 regularization when the user specifies a value other than `None` for the regularization parameter. Ideally, you do not implement any new functions, and only modify existing ones to add this option. (4 points)

(iii) For the breast cancer data, we fit 3 models with regularization strength $\lambda = 0, 0.01, 0.2$. Based on their validation accuracies which one would you prefer? Report the testing accuracy for this model by modifying the line in the `main` function to set the best model appropriately. (3 points)

(i) See code submission.
(ii) See code submission.
(iii) 1. Simulated data accuracy: 0.8935

2. lamda = 0.0: 0.95604395

3. lamda = 0.1: 0.967032967

4. lamda = 0.2: 0.95604395

5. Testing accuracy = 0.9649122807

Based on the validation accuracies, I would prefer the model with a regularization strength of $\lambda = 0.01$.

[1] Nuclear Feature Extraction For Breast Tumor Diagnosis. Street et al (1992).

(iv) Though the functionality for cleaning and processing the breast cancer data has been provided to you in `data_loader.py`, it's important you become comfortable with this pipeline, and are able to implement your own for the final project (and beyond.) In your own words, describe the processing steps performed by the `load_breast_cancer_data` function. Hint: also examine the original breast cancer data file to see if any features are dropped at processing time. (5 points)

The `load_breast_cancer_data` function loads the csv file and assigns 0 and 1 labels to represent either malignant or benign assignments rather than the character letter "M". This step helps us work with a binary classifier. The function also splits up each data entry into a matrix containing the specific features that are being classified for each entry. The data is split into training, validation, and test sets, for evaluation purposes. The data is then standardized by applying the standard deviation method. Standardizing data is important because it helps create a uniform analysis of the data and gets rid of any extraneous data points that may deter the model's performance. The final values returned from the function are the training, validation, and test sets. Overall, this function serves to clean, process, and create structure to the data before it is used to test the model. This allows the model to know what to expect so it is not thrown off by missing or erroneously-formatted data. The model now knows what to expect as test inputs.

It was very interesting to evaluate the `breast_cancer`.csv file as it contains lots of different descriptive identifiers for tumors such as texture_worst, smoothness_worst, fractal_dimension, among others.

It appears that potentially only features that contain 'mean' in the name were used in processing. This eliminates a lot of the other data that was used for model analysis. Such examples of features with 'mean' in the feature name include: radius_mean, texture_mean, smoothness_mean, and others.

Important submission note:
Report the validation accuracies for the breast cancer data, the testing accuracy of the final model, and answers to the above questions in the PDF that you turn in. Hand in your Python code separately on Gradescope under HW 2 Programming. Please do not modify the existing code in the skeleton, except for the one line in the main function. Please make sure your variables and functions are aptly named, and your code is clear and readable, with comments where appropriate.