6.390: Midterm Exam, Spring 2023

Do not tear exam booklet apart!

- This is a closed book exam. One page (8 1/2 in. by 11 in.) of notes, front and back, are permitted. Calculators are not permitted.
- The total exam time is 2 hours.
- The problems are not necessarily in any order of difficulty.
- Record all your answers in the places provided. If you run out of room for an answer, continue on a blank page and mark it clearly.
- If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.
- If you absolutely *have* to ask a question, come to the front.
- **Write your name on every piece of paper.**

Name: _____ MIT Email: _____

Question	Points	Score
1	19	
2	12	
3	19	
4	12	
5	18	
6	11	
7	9	
Total:	100	

Name: _____

Regression or Progression?

1. (19 points) Suppose that you are given a small dataset and you would like to learn the parameters of a linear regressor hypothesis taking the form $h(x) = \theta^\top x + \theta_0$ for fitting the data.

   (a) First, consider the following dataset $\mathcal{D}_1$ containing three feature-label pairs:

x	y
-4	3
2	-1
-1	0

   Suppose that you would like to minimize the following objective function:

   $$J_1(\theta, \theta_0) = \frac{1}{n}\sum_{i=1}^{n}(\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2.$$

   You decide that you would like to find the analytic solution. You start by defining $Y = [y^{(1)},\ y^{(2)}, \ldots,\ y^{(n)}]^\top$ and $\bar{\theta} = [\theta,\ \theta_0]^\top$ and rewrite your objective function as,

   $$J_2(\bar{\theta}) = \frac{1}{n}(\tilde{X}\bar{\theta} - Y)^\top(\tilde{X}\bar{\theta} - Y).$$

   i. What is the data matrix $\tilde{X}$ with data points as rows corresponding to the dataset $\mathcal{D}_1$ which ensures that the two objective functions J_1, J_2 are equivalent?

   $\tilde{X} =$

   ii. Does J_2 have an analytic solution for dataset $\mathcal{D}_1$? Explain your answer. (Note: you do not need to compute the analytic solution.)

   Circle one: Yes No

   Explanation:

Name: _____

(b) As your dataset is quite small, you decide to add a second feature for each of the three datapoints. Consider the new dataset $\mathcal{D}_2$ defined as:

x_1	x_2	y
-4	-8	3
2	4	-1
-1	-2	0

i. Does J_2 have an analytic solution for dataset $\mathcal{D}_2$? Explain your answer. (Note: you do not need to compute the analytic solution.)

Circle one: Yes No

Explanation:

ii. Consider a new objective function with an added regularization term:

$$J_3(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} (\theta^\top x^{(i)} + \theta_0 - y^{(i)})^2 + 0.1(\|\theta\|^2 + \theta_0^2)$$

Will adding this particular form of regularization in J_3 improve the generalization of our model? Explain your answer.

Circle one: Yes No

Explanation:

iii. Does J_3 have an analytic solution for dataset $\mathcal{D}_2$? Explain your answer. (Note: you do not need to compute the analytic solution.)

Circle one: Yes No

Explanation:

(c) Consider a new dataset $\mathcal{D}_3$ which includes a fourth data point:

x_1	x_2	y
-4	-8	3
2	4	-1
-1	-2	0
5	8	-4

Does J_2 have an analytic solution for dataset $\mathcal{D}_3$? Explain your answer. (Note: you do not need to compute the analytic solution.)

Circle one: Yes No

Explanation:

Name: _____

Discerning Descents

2. (12 points) Indicate whether the following statements are true or false. Explain your answer.

   (a) Gradient descent is sure to converge, to some value, for any step size greater than 0.

   Circle one: True False

   Explanation:

   (b) Stochastic gradient descent reduces the objective function value at every iteration.

   Circle one: True False

   Explanation:

   (c) When the algorithms converge, stochastic gradient descent always finds the same solution as gradient descent.

   Circle one: True False

   Explanation:

   (d) The more features that we use to represent our data, the better the learning algorithm will generalize to new data points.

   Circle one: True False

   Explanation:

Name: _____

Logistic Regression Logistics

3. (19 points) You would like to train a binary linear logistic classifier of the form,

$$h(x; \theta, \theta_0) = \sigma(\theta^T x + \theta_0),$$

where $\sigma(z) = 1/(1 + e^{-z})$ is the sigmoid function, by minimizing an objective function,

$$J(\theta, \theta_0) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(g^{(i)}, a^{(i)}) + \lambda \|\theta\|^2,$$

where $g^{(i)} = h(x^{(i)}; \theta, \theta_0)$ is the "guess" or model output and $a^{(i)} = y^{(i)}$ is the "actual" label.

(a) Typically, we use the negative log-likelihood (NLL) as the loss function for classification,

$$\mathcal{L}_{\text{NLL}}(g, a) = -(a \log g + (1 - a) \log(1 - g)).$$

Additionally, recall that

$$\frac{\partial \mathcal{L}_{\text{NLL}}(g, a)}{\partial g} = \frac{1 - a}{1 - g} - \frac{a}{g}.$$

We could also consider using squared loss for classification,

$$\mathcal{L}_{\text{squared}}(g, a) = (g - a)^2.$$

i. Consider the case when our guess is $g \approx 0$ but the actual value should be $a = 1$. Compute the following values:

$\mathcal{L}_{\text{NLL}}(g, a) \approx$ _____

$\mathcal{L}_{\text{squared}}(g, a) \approx$ _____

$\partial \mathcal{L}_{\text{NLL}}(g, a)/\partial g \approx$ _____

$\partial \mathcal{L}_{\text{squared}}(g, a)/\partial g \approx$ _____

ii. Now, let's instead assume that $g = 0.5$ and the actual label is still $a = 1$. Compute the following values:

$\partial \mathcal{L}_{\text{NLL}}(g, a)/\partial g =$_____

$\partial \mathcal{L}_{\text{squared}}(g, a)/\partial g =$ _____

iii. Is it ever possible to obtain a guess of $g = 1$? Explain why or why not.

Circle one: Yes No

Explanation:

iv. Based on these observations, briefly explain why we might prefer one of the loss functions presented above over the other when training a linear logistic classifier.

Circle one: Yes No

Explanation:

(b) You decide to train your binary linear logistic classifier by minimizing the J objective function with the NLL loss function by using gradient descent. A few of your friends comment on your methodology.

i. Chris thinks that sigmoids are too confusing! He suggests that you can replace the sigmoid function with a step function for training your model.

Does Chris's scheme make sense? Explain your answer.

Circle one: Yes No

Explanation:

ii. Jojo claims that you can generalize from binary to trinary classification as follows: let the hypothesis take the form $h_3(x;\theta,\theta_0) = 3\sigma(\theta^\top x + \theta_0)$, and learn the parameters θ, θ_0 by minimizing the same objective J with NLL loss, with class labels $0, 1, 2$. Then, you can assign classes using the output of your hypothesis as,

$$\begin{aligned} &\text{class } 0, && \text{if } 2 \leq h_3(x), \\ &\text{class } 1, && \text{if } 1 \leq h_3(x) < 2, \\ &\text{class } 2, && \text{if } h_3(x) < 1. \end{aligned}$$

Does Jojo's scheme make sense? Explain your answer.

Circle one: Yes No

Explanation:

iii. Mona overhears your conversation with Jojo and suggests that instead of using a single classifier, you learn two different binary linear logistic classifiers.
First, learn one binary classifier h_{01} by minimizing J with the NLL loss function on a dataset where class 0 is labeled 0 and class 1 is labeled 1, and class 2 feature vectors are omitted.
Then, learn a second classifier h_{12} by minimizing J with the NLL loss function on a dataset where class 0 feature vectors are omitted, class 1 is labeled 0, and class 2 is labeled 1.
You can assign classes using the output of your two hypotheses by performing the following sequentially:

$$\begin{aligned} \text{if } h_{01}(x) < 0.5, &\quad \text{assign class } 0, \\ \text{else, if } h_{12}(x) < 0.5, &\quad \text{assign class } 1, \\ \text{otherwise}, &\quad \text{assign class } 2. \end{aligned}$$

Does Mona's scheme make sense? Explain your answer.

Circle one: Yes No

Explanation:

Name: _____

Won't You Be My Neighbor?

4. (12 points) Indicate whether the following statements are true or false. Explain your answer.

   (a) Consider a classification problem where the training dataset consists of n data points that all have different feature values.

      i. A k-nearest-neighbor classifier with $k = 1$ will always have 100% training accuracy on this dataset.

      Circle one: True False

      Explanation:

      ii. A k-nearest-neighbor classifier with $k > 1$ will always have 100% training accuracy on this dataset.

      Circle one: True False

      Explanation:

      iii. In general, using a k-nearest-neighbors classifier with $k > 1$ as opposed to 1-nearest-neighbor can effectively reduce the tendency of the model to overfit to training data.

      Circle one: True False

      Explanation:

(b) Consider a regression problem where the training dataset consists of n data points that all have different feature values.

i. The MSE (mean squared error) of a k-nearest-neighbor regressor with $k = 1$ will always be 0 on this dataset.

Circle one: True False

Explanation:

ii. The MSE (mean squared error) of a k-nearest-neighbor regressor with $k > 1$ will always be 0 on this dataset.

Circle one: True False

Explanation:

iii. A k-nearest-neighbor regressor with $k = n$ trained on this dataset will always output constant prediction.

Circle one: True False

Explanation:

Name: _____

The Rise of ELU

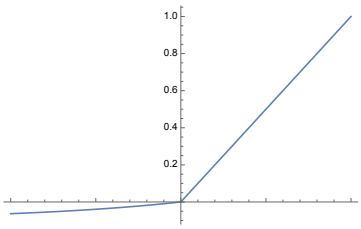
5. (18 points) We have been considering the ReLU function for an internal non-linearity in neural networks:

$$\text{ReLU}(z) = \begin{cases} z & \text{if } z > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Let's also consider the ELU (exponential linear unit) function:

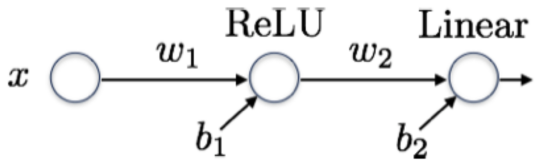
$$\text{ELU}_\alpha(z) = \begin{cases} z & \text{if } z > 0, \\ \alpha(e^z - 1) & \text{otherwise,} \end{cases}$$

which is shown here for $\alpha = 0.1$:



(a) What is $\partial \text{ELU}_\alpha(z)/\partial z$?

(b) We have the simple neural network shown below:



We will train the neural network using the squared loss function,

$$\mathcal{L}_{\text{squared}}(g, a) = (g - a)^2.$$

Our target output is 1 and the initial weights (including the offsets b_i) are -0.1.

i. Assume the input is $x = 1$. What is the sign of the update that will be applied to w_1, using a step size of 1? Explain your answer.

ii. Assume that instead of the ReLU unit we have an **ELU** unit. What is the sign of the gradient update that will be applied to w_1, using a step size of 1? Explain your answer.

iii. Assume we initialize our networks with weights drawn uniformly from the range $[-0.1, 0.1]$ What advantage does using **ELU** have over **ReLU** for use in a multi-layer neural network?

iv. What advantage does using **ELU** have over the **linear** activation function for use in the internal units of a multi-layer neural network?

Name: _____

Classification (feat. Features)

6. (11 points) Here is a training data-set of four data points (with 1-dimensional feature x) for a classification problem:

x	y
-1	-1
0	$+1$
1	$+1$
2	-1

(a) What are the feature values under the feature transformation $\phi(x) = [x, x^2, (x-1)^2]^\top$ for each of these four data points?

(b) The given data set is linearly separable in the new feature space $\phi(x) = [x, x^2, (x-1)^2]^\top$. Suppose we have a linear classifier with $\theta = [-1, -1, -1]^\top$ and an unknown θ_0. Provide a value of θ_0 that results in a correct separator on the training data. Justify your answer.

$\theta_0 =$

(c) Suppose we use a linear classifier $\theta = [-1, -1, -1]^\top$ and $\theta_0 = 4$ in the new feature space $\phi(x) = [x, x^2, (x-1)^2]^\top$. For what range of values in the original x feature space would a data point be classified as positive?

Name: _____

Detective on the Case

7. (9 points) As a machine learning expert, your friends often come to you looking for advice on how to fix their code. For each of the following snippets of pseudocode, identify which line(s) of code have errors and explain how you would change them.

   Here are five commonly referenced functions. You do not need to debug their contents. You can assume that they operate correctly.

```
def objective_function(feature_vectors, labels, model, hyperparams=None)
    # Defines the objective function
    return objective_function

def objective_value(feature_vectors, labels, model, hyperparams=None ...
                    objective_fn=objective_function):
    # Computes the objective function value
    return objective_value

def train_model(feature_vectors, labels, hyperparams=None, ...
                    objective_fn=objective_function):
    # Learns a model using some algorithm on the dataset
    # e.g., by minimizing an objective function
    return model

def load_dataset(fname):
    # Loads the dataset
    return feature_vectors, labels

def train_val_split(feature_vectors, labels):
    # splits the given dataset into two datasets
    # e.g., for training and validating a model
    return train_vectors, train_labels, val_vectors, val_labels
```

   (a) Find the error(s) in this pseudocode for training and validating a model:

```
1.  x_full, y_full = load_dataset("data.csv")
2.  my_model = train_model(x_full, y_full)
3.  x_train, y_train, x_val, y_val = train_val_split(x_full, y_full)
4.  error = objective_value(x_val, y_val, my_model)

    print("My model's performance: " + str(error))
```

   Line(s) of code with error:

   Explanation of fixing the error:

(b) Find the error(s) in this pseudocode for training and testing a regularized model:

```
1.  lam = 0.01
2.  x_full, y_full = load_dataset("data.csv")
3.  x_train, y_train, x_val, y_val = train_val_split(x_full, y_full)
4.  my_model = train_model(x_train, y_train, hyperparams=lam)
5.  error = objective_value(x_val, y_val, my_model, hyperparams=lam)

print("My regularized model's performance: " + str(error))
```

Line(s) of code with error:

Explanation of fixing the error:

(c) Find the error(s) in this pseudocode for shipping a regularized model with a validated hyperpameter value:

```
1.  x_full, y_full = load_dataset("data.csv")
2.  x_train, y_train, x_val, y_val = train_val_split(x_full, y_full)

3.  lambda_values = [0.0001, 0.001, 0.01, 0.1, 1.0]
4.  min_error = 1e10 # assume that any model will have error < 1e10

5.  for lam in lambda_values:
6.      model = train_model(x_train, y_train, hyperparams=lam)
7.      error = objective_value(x_val, y_val, model)
8.      if error < min_error:
9.          lambda_star = lam
10.         min_error = error

11. best_model = train_model(x_train, y_train, hyperparams=lambda_star)

print("Finished training validated model. Ready for testing!")
```

Line(s) of code with error:

Explanation of fixing the error:

Name: _____

Work space

Name: _____

Work space