**CS4851/6851 IDL: Homework 1** **January 27, 2022**

Note: All coding problems are to be implemented in a single jupyter notebook.

Note: this is the distribution of questions:

1. Question 1 to Question 8: Required for everyone.
2. Question 9 - Question 10: Required only for Graduate Students
3. Question 11 - Question 12: Bonus marks

**Problem 1 (5 points)**

What is the difference between Root Mean Squared Error (RMSE) and Mean Squared Error (MSE). Describe the context in which they will be most useful? Which of these losses penalizes large differences between the predicted and expected results and why?

* Root Mean Squared Error (RMSE) defines how close a fitted line is to data points. The square root of the average squared difference between the predicted and actual values in a dataset. The lower the RMSE the better the model fits
* Mean Squared Error (MSE) defines the average squared difference between the predicted and actual values in a dataset. The lower the MSE, the better the model fits

**Problem 2 (5 points)**

Which of the following statements is true about the relationship between features and dataset sizes? Choose one of the following and explain the chosen answer

1. When training a model, as you add more features to the dataset, you often need to increase the dataset’s size to ensure the model learns reliably.
2. When training a model, adding more features to the dataset increases the amount of information you can extract from the dataset. This allows you to use smaller datasets to train the model and still achieve good generalization performances from the training data.
3. When training a learning algorithm, as you decrease the number of features in the dataset, you need to increase the training sample size to make up the difference.
4. When training a learning algorithm, the number of features in your dataset is entirely dependent on the amount of information you can extract from the dataset. **Answer:** More features does not necessarily mean better model results, too many features can lead to a hinderance due to the unneeded data causing the model to be slower than normal

1

**Problem 3 (4 points)**

You are building a deep learning model and after *X* epochs you see that the training loss is decreasing, while your validation loss is either constant or in-creasing. What would be the most likely cause of this and suggestion to rectify it

1. Learning rate is too high
2. Gradient descent does not converge
3. Insuﬀicient training data size
4. Gradient descent is stuck in a local minimum

**Problem 4 (4 points)**

Perceptron

1. is a linear classifier. A. True B. False
2. and cannot be trained on linearly unseperable data. A. True B. False

**Problem 5 (4 points)**

Logistic Regression

1. is a linear classifier. A. True B. False
2. and always has a unique solution A. True B. False

**Problem 6 (10 points)**

Which of the following is true, given the optimal learning rate?

1. Batch gradient descent is always guaranteed to converge to the global opti-mum of a loss function.
2. Stochastic gradient descent is always guaranteed to converge to the global optimum of a loss function.

2

1. For convex loss functions (i.e. with a bowl shape1), batch gradient descent is guaranteed to eventually converge to the global optimum while stochastic gradient descent is not.
2. For convex loss functions (i.e. with a bowl shape), stochastic gradient de-scent is guaranteed to eventually converge to the global optimum while batch gradient descent is not.
3. For convex loss functions (i.e. with a bowl shape), both stochastic gradient descent and batch gradient descent will eventually converge to the global optimum.
4. For convex loss functions (i.e. with a bowl shape), neither stochastic gradient descent nor batch gradient descent are guaranteed to converge to the global optimum.

**Problem 7 (5 points)**

Given the following data on a 2D plane :

x y 

-1 -2

1 -1

* 3

Fit a linear regression model to the data without the constant term: *yi* = *βxi* + *ϵi*. Give an expression of the minimization problem for finding *β*, Showhow to compute it’s value and show the value

Yi is the dependent variable, xi is the independent variable, €I is the error between the observed yi and what the model predicts and β is slope parameter.

Model describes the relation between xi and yi using the slope parameter.

The error €I = yi – βxi.

|  |  |  |  |
| --- | --- | --- | --- |
| X | Y | XY | X2 |
| -1 | -2 | 2 | 1 |
| 1 | -1 | -1 | 1 |
| 2 | 3 | 6 | 4 |
| 2 | 0 | 7 | 6 |

Least square error β

Q = -β)2, = = 0, - β.

**Problem 8 (20 points)**

Perform a polynomial fitting to compute a design matrix *X* such that:

|  |  |
| --- | --- |
| *Xij* = *yij* | (1) |

You should implement this **without a single for loop** by using vectorization and broadcast. Here (1 *≤ j ≤* 50) and *y* = *{−*20*, −*19*.*9*, ...,* 20*}*. Implement code that generates such a matrix.

1More formally *f* is convex on [*a, b*] if *∀x*1*, x*2 *∈* [*a, b*] if *f* (*λx*1 + (1 *− λ*)*x*2) *≤ λf* (*x*1) + (1 *− λ*)*f* (*x*2)

3

Bonus for undergraduates beyond this line

**Problem 9 (15 points)**

Is this a good idea to initialize parameters with zeros when training a logistic regression model? Explain your answer. No because after backpropagation each neural network will have the same weight.

**Problem 10 (15 points)**

As discussed in the class, the logistic regression likelihood simplifies to the fol-lowing form:

cost(*f* (*x*)*, y*) = *−y* log(*f* (*x*)) *−* (1 *− y*) log(1 *− f* (*x*))

* *∈ {*0*,* 1*}* class label

What will happen when *y* = 1 and *f* (*x*) = 1? Will this work in actual implementations?

When y = 1 and f(x) = 1, the equation will undefined.

Extra credit problems

**Problem 11 (extra credit 20 points)**

In the three examples of Figure , find the perceptron weights *w*0*, w*1*, w*2 for each of them. The line that separates the two classes is given by the decision boundary that divided into positive and negative classes. (No calculations required; you can simply note the answer.)

A picture containing object, antenna

Description automatically generated

Figure 1: Binary class data and separating decision boundaries.

4

**Problem 12 (extra credit 20 points)**

Consider function *E* below:

*E* = *g*(*x, y, z*) = *σ*(*c*(*ax* + *by*) + *dz*)*,*

where *x, y, z* are input variables and *a, b, c, d* are parameters. *σ* is the logistic

|  |  |  |
| --- | --- | --- |
| sigmoid function defined as: |  |  |
| *σ*(*z*) = | 1 | (2) |
|  |
| 1 + *e−z* |

Note that *E* is also a function of *a, b, c, d* and for completeness could be written as *E*(*x, y, z, a, d, c, d*).

Derive expressions of partial derivatives of *E* with respect to parameters *a, b, d*

i.e *∂E∂a , ∂E∂b , ∂E∂d*

5