CS 615 - Deep Learning

Assignment 1 - Forward Propagation
Winter 2022

Introduction

In this first assignment we'll begin designing and implementing the modules we'll use in our deep learning architectures. In addition, we'll start getting used to importing datasets.

Allowable Libraries/Functions

Recall that you **cannot** use any ML functions to do the training or evaluation for you. Using basic statistical and linear algebra function like *mean*, *std*, *cov* etc.. is fine, but using ones like *train* are not. Using any ML-related functions, may result in a **zero** for the programming component. In general, use the "spirit of the assignment" (where we're implementing things from scratch) as your guide, but if you want clarification on if can use a particular function, DM the professor on discord.

Grading

Theory	15pts
Implementation of layers w/ forward propagation	50pts
Basic test on connected layer set	20pts
Test with medical cost dataset on connected layer set	15pts
TOTAL	100pts

Table 1: Grading Rubric

Datasets

Medical Cost Personal Dataset (mcpd_augmented.csv)   This dataset consists of data for 1338 people in a CSV file. The original dataset, found at https://www.kaggle.com/mirichoi0218/insurance, contains the following information for each person:

1. age

2. sex

3. bmi

4. children

5. smoker

6. region

7. charges

For the purposes of this assignment, we have converted the *sex* and *smoker* features into binary features and the *region* into an enumerated feature $\in \{0, 1, 2, 3\}$. In addition, we omitted the *charges* information (in a subsequent assignment we will look to predict this).

1 Theory

1. Given a single input observation $x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ and a fully connected layer with weights of $W = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ as biases $b = \begin{bmatrix} -1 & 2 \end{bmatrix}$, what are the output of the fully connected layer given x as its input (5pts)?

2. Given an input, $h = [10, -1]$, what would be the output(s) if this data was processed by the following activation functions/layers (10pts)?

   (a) Linear

   (b) ReLu

   (c) Sigmoid

   (d) Hyperbolic Tangent

   (e) Softmax

2   ABC and Input Layer

Referring to the lecture slides, first implement an *abstract base class* called *Layer* that has the following:

- A constructor that initializes the attributes, one for the previous input and one for the previous output, to empty lists.

- Getter and setter methods for the attributes.

- An abstract method for forward propagation, *forward.*

- An abstract method for getting the layer's gradient, *gradient.*

Next, implement a class called *InputLayer* that inherits from your abstract base class. This class' constructor should take as a parameter your dataset, and initialize two attributes, *meanX* and *stdX* to be row vectors of the mean and standard deviation, respectively, of the features of your dataset. For numeric stability, set any feature that has a standard deviation of zero to 1. In addition, you must implement the abstract method *forward* such that it takes a data matrix, X, as a parameter, sets the parent class' previous input attribute to it, and computes the *zscored* version of this data using the *meanX* and *stdX* attributes, setting the parent class' previous output to this and returning this . You'll also need to implement the abstract method *gradient* as well, but it can just return nothing (for now!).

3   Activation Layers

Nex,t implement classes for the following activation functions:

- *ReLuLayer*

- *SigmoidLayer*

- *SoftmaxLayer*

- *TanhLayer* (Hyperbolic Tangent Function)

Each class should inherit from the *Layer* abstract base class, and must implement the *forward* method that takes in data as a parameter, sets its parent class' previous input to that data, computes the output values, setting the parent class' previous output to that, and returns that output.

4   FullyConnectedLayer

Finally, let's create a class for a fully connected layer, aptly called *FullyConnected.* This too should inherit from *Layer.* Following the material in the lecture slides, this class should have two attributes, a weight matrix and a bias vector. The constructor should take in two parameters, the number of features coming in, and the number of features coming out of this layer, and use these to initialize the weights to be random values in the range of $\pm 10^{-4}$. Its *forward* method once again takes in data X, storing it in its parent's previous input attribute, and computes the output (storing it with the parent class, and returning it) as:

$$Y = XW + b$$

where W is the weight matrix and b is the bias vector. For now, the *gradient* method may return nothing. In addition provide getter and setter methods for the weight and bias attributes.

5 Testing the layers

Let's test each layer using the following input data (this can be considered two observations, with four features per observation):

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

For reproducability, seed your random number generator to zero prior to running your tests.

In your report provide:

- Output of input layer

- Output of fully connected layer (output is of size two)

- Output of ReLu activation layer

- Output of sigmoid activation layer

- Output of softmax activation layer

- Output of tanh activation layer

6 Connecting Layers and Forward Propagate

Now let's assemble a simple network and forward propagate data through it!

Our architecture will be:

$$\text{Input} \rightarrow \text{FC (2 outputs)} \rightarrow \text{Sigmoid}$$

We'll once again use the data X from our prior problem as the input to this pipeline. From an implementation standpoint, you'll likely want to create instances of your classes and organize them in some sort of an ordered structure such that the output of one layer is the input of the next.

In your report provide the output from each layer.

7 Testing on full dataset

To test your implementation on a real dataset, we'll use the augmented medical cost dataset mentioned earlier in the assignment. Read in the dataset as your input data X and pass it through the architecture from the previous problem.

In your report, just provide the output of the last layer pertaining to the **first observation**.

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF Writeup

2. Source Code

3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment.

The PDF document should contain the following:

1. Part 1:

   (a) Your solutions to the theory question

2. Parts 2-4: Nothing

3. Part 5:

   (a) The output for each layer when using the provided X as its input.

4. Part 6:

   (a) The output of the final layer, when given input X as the input to the first layer.

5. Part 7:

   (a) The output pertaining to the first observation from the final layer, when given the augemented medical cost dataset as its input.