CS 615 - Deep Learning

Assignment 3 - Backprop and Basic Architectures
Winter 2022
By: Willie Hood

02/07/22

1 Theory

1. For the function $J = (x_1 w_1 - 5x_2 w_2 - 2)^2$, where $w = [w_1, w_2]^T$ are our weights to learn:

   (a) What are the partial gradients, $\frac{\partial J}{\partial w_1}$ and $\frac{\partial J}{\partial w_2}$? Show work to support your answer (6pts).
   ANSWER:

$$\frac{\partial J}{\partial w_1} = 2(x_1 w_1 - 5x_2 w_2 - 2)(x_1)$$

$$\frac{\partial J}{\partial w_2} = 2(x_1 w_1 - 5x_2 w_2 - 2)(-5x_2)$$

$$\boxed{\nabla J(w_1, w_2) = < 2x_1(x_1 w_1 - 5x_2 w_2 - 2), -10x_2(x_1 w_1 - 5x_2 w_2 - 2) >}$$

   (b) What are the value of the partial gradients, given current values of $w = [0, 0]^T, x = [1, 1]^T$ (4pts)?
   ANSWER:

$$\boxed{\nabla J(0, 0) = < -4, 20 >}$$

2. Given the objective function $J = \frac{1}{4}(x_1 w_1)^4 - \frac{4}{3}(x_1 w_1)^3 + \frac{3}{2}(x_1 w_1)^2$:

   (a) What is the gradient $\frac{\partial J}{\partial w_1}$ (2pts)?
   ANSWER:

$$\frac{\partial J}{\partial w_1} = \frac{4}{4}(x_1 w_1)^3 (x_1) - \frac{12}{3}(x_1 w_1)^2 (x_1) + \frac{6}{2}(x_1 w_1)$$

$$\rightarrow x_1(x_1 w_1)^3 - 4x_1(x_1 w_1)^2 + 3x_1(x_1 w_1)$$

$$\boxed{\nabla J(w_1) = < x_1(x_1 w_1)^3 - 4x_1(x_1 w_1)^2 + 3x_1(x_1 w_1) >}$$

   (b) What are the locations of the extrema points for this objective function J if $x_1 = 1$? Recall that to find these you set your equation to zero and solve for, in this case, w_1. (5pts)
   ANSWER:

$$0 = w_1^3 - 4w_1^2 + 3w_1$$

$$0 = (w_1)(w_1^2 - 4w_1 + 3)$$

$$0 = (w_1)(w_1 - 3)(w_1 - 1)$$

$$\boxed{\mathrm{w}_1 = 0, 3, 1}$$

   (c) What does J evaluate to at each of your extrema points, again when $x_1 = 1$ (3pts)?
   ANSWER:

$$@x_1 = 1$$

$$@w = 3$$

$$J = \frac{1}{4}3^4 - \frac{4}{3}3^3 + \frac{3}{2}3^2 = \boxed{\text{-2.25}}$$

$$@w = 1$$

$$J = \frac{1}{4} - \frac{4}{3} + \frac{3}{2} = \boxed{0.42}$$

$$@w = 0$$

$$J = \boxed{0}$$

2 Visualizing Gradient Descent

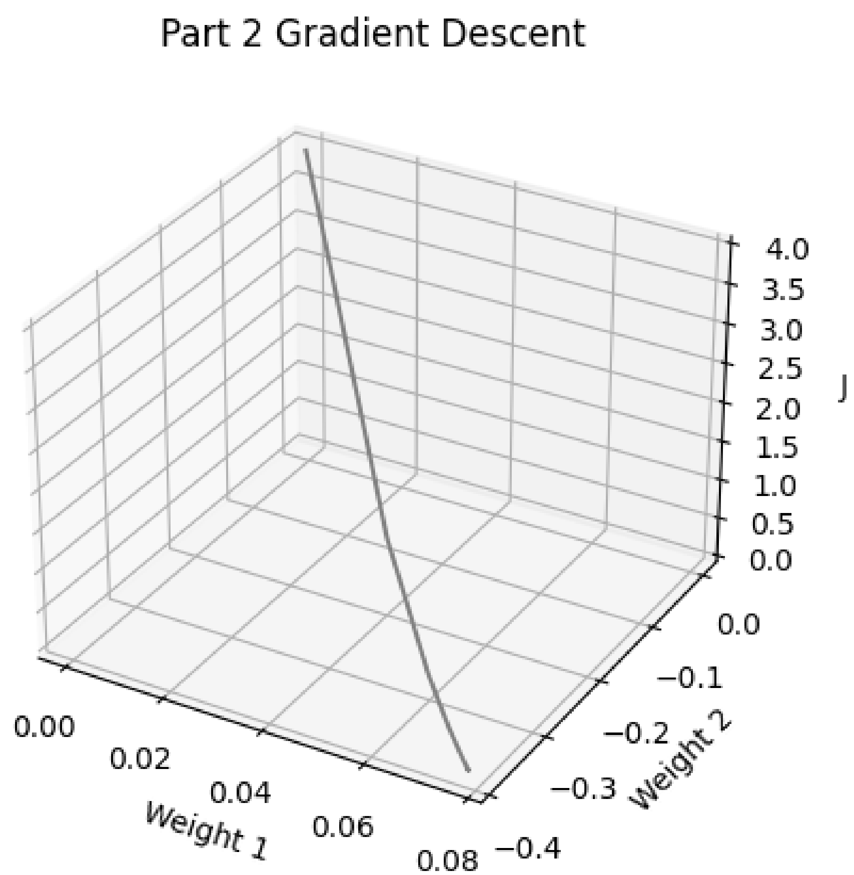
In this section we want to visualize the gradient descent process for the following function (which was part of one of the theory questions):

$$J = (x_1 w_1 - 5x_2 w_2 - 2)^2$$

Hyperparameter choices will be as follows:

- Initialize your parameters to zero.
- Set the learning rate to $\eta = 0.01$.
- Terminate after 100 epochs.

Using the partial gradients you computed in the theory question, perform gradient descent, using $x = [1, 1]^T$. After each training epoch, evaluate J so that you can plot w_1 vs w_2, vs J as a 3D line plot. Put this figure in your report.



4 Linear Regression

In this section you'll use your modules to assemble a linear regression model and train and validate it using the *medical cost dataset*. The architecture of your linear regression should be as follows:

$$Input \rightarrow \text{Fully-Connected} \rightarrow \text{Least-Squared-Objective}$$

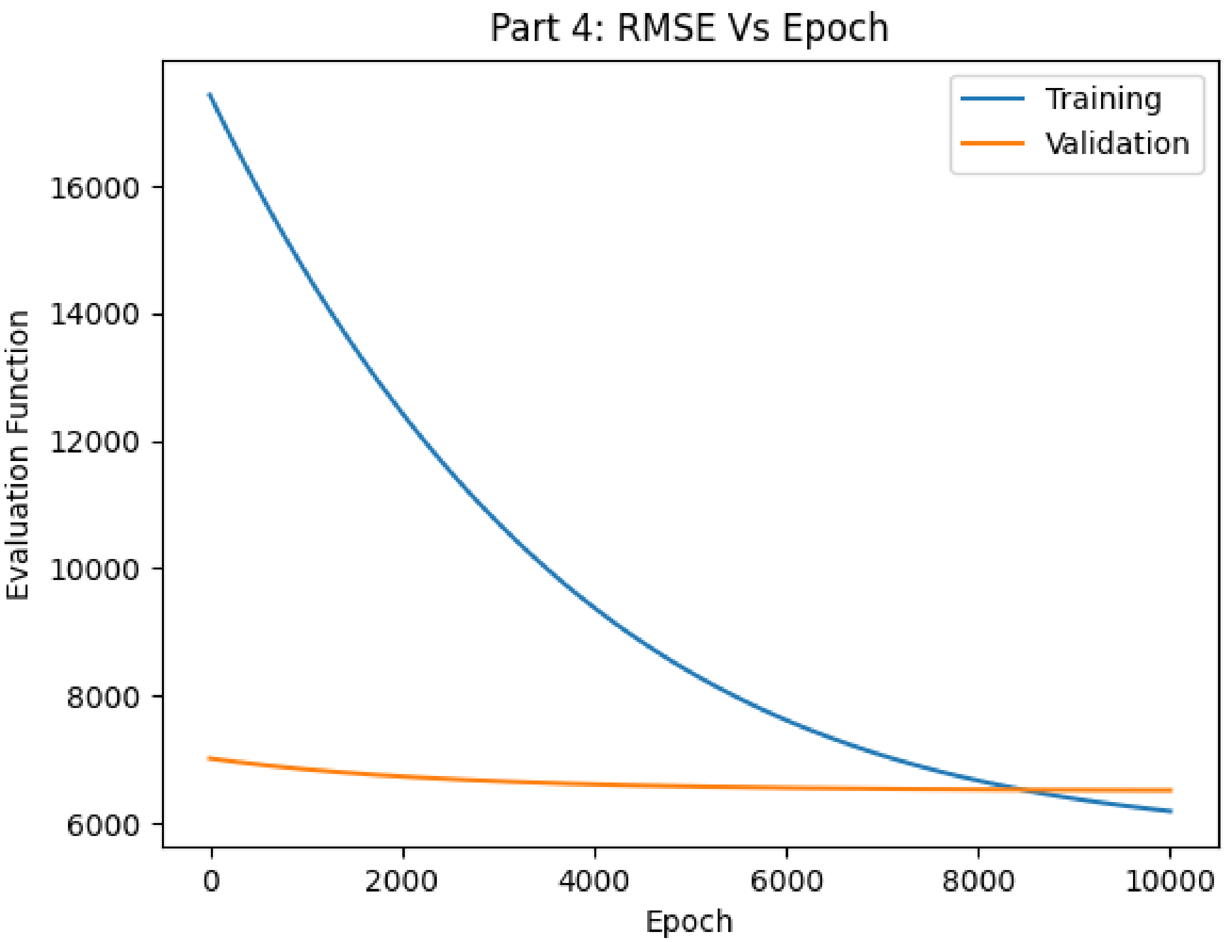
Your code should do the following:

1. Read in the dataset to assemble X and Y
2. *Shuffle* the dataset, extracting $\frac{2}{3}$ for training and $\frac{1}{3}$ for validation.
3. Train, via gradient learning, your linear regression system using the training data. Refer to the pseudocode in the lecture slides on how this training loop should look. Initialize your weights to be random values in the range of $\pm 10^{-4}$ and your learning rate to be $\eta = 10^{-4}$. Terminate the learning process when the absolute change in the mean absolution percent error (MAPE) on the training data is less than 10^{-10} or you pass $10,000$ epochs. During training, keep track of the RMSE and MAPE for both the training and validation sets so that we can plot these as a function of the epoch.

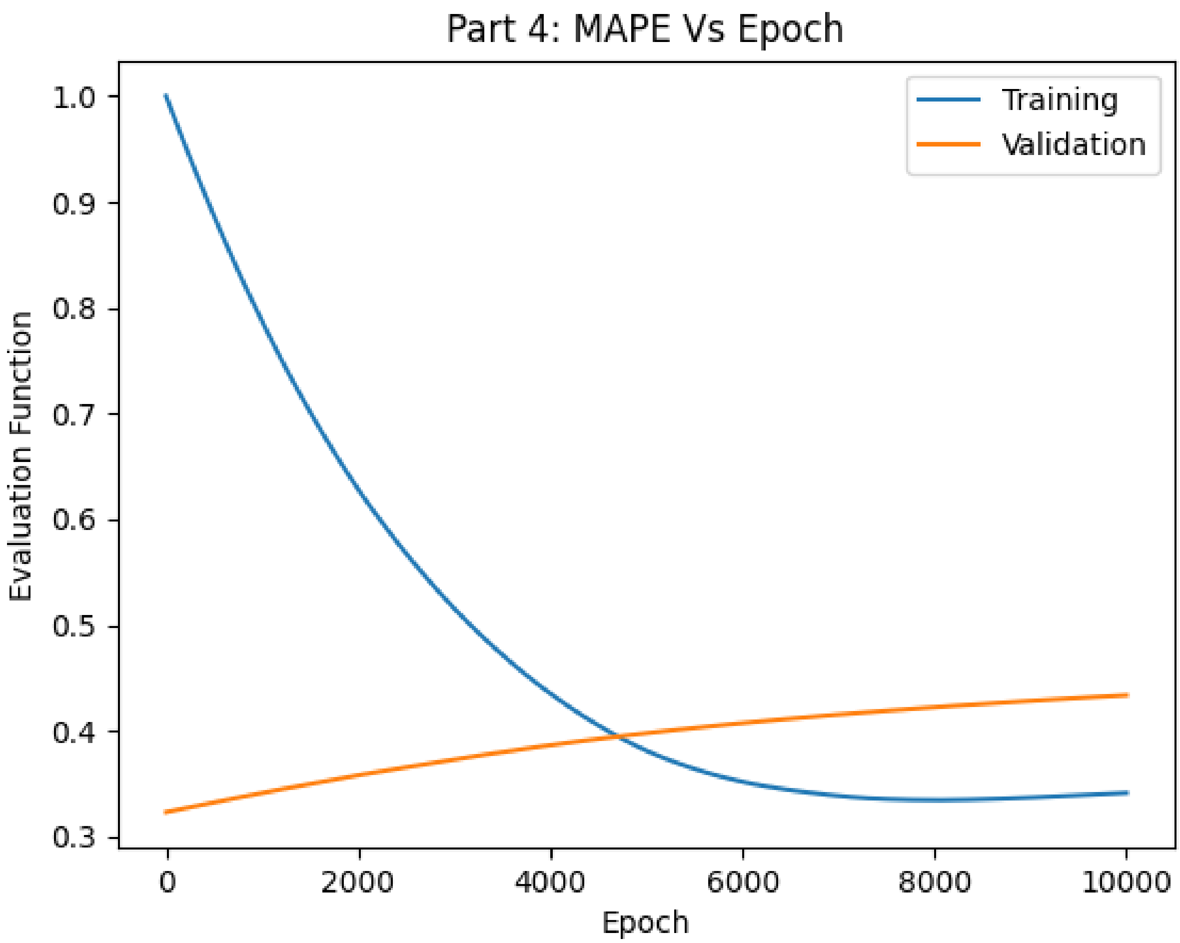
In your report provide:

1. Your plots of training and validation RMSE vs epoch.
2. Your plots of training and validation MAPE vs epoch.

GRAPHS:



Part 4: RMSE Vs Epoch
Training
Validation
16000
14000
12000
10000
8000
6000
Evaluation Function
0
2000
4000
6000
8000
10000
Epoch



Part 4: MAPE Vs Epoch
Training
Validation
Evaluation Function
Epoch
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2000
4000
6000
8000
10000

5 Logistic Regression

Next we'll use a logistic regression model on the *kid creative* dataset to predict if a user will purchase a product. The architecture of this model should be:

$$Input \rightarrow \text{Fully-Connected} \rightarrow \text{Sigmoid-Activation} \rightarrow \text{Log-Loss-Objective}$$

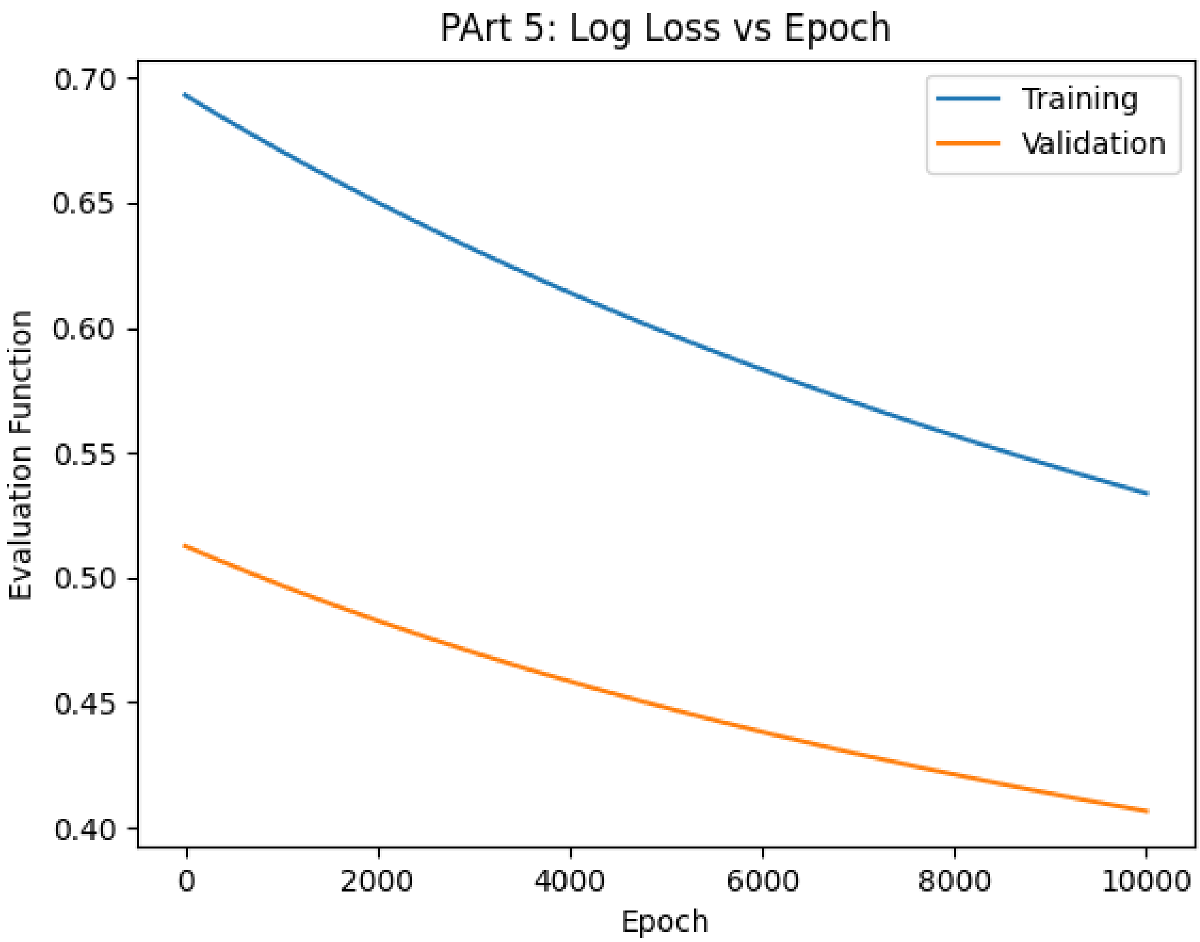
Your code should do the following:

1. Read in the dataset to assemble X and Y
2. *Shuffle* the dataset, extracting $\frac{2}{3}$ for training and $\frac{1}{3}$ for validation.
3. Train, via gradient learning, your logistic regression system using the training data. Initialize your weights to be random values in the range of $\pm 10^{-4}$ and your learning rate to be $\eta = 10^{-4}$. Terminate the learning process when the absolute change in the log loss is less than 10^{-10} or you pass $10,000$ epochs. During training, keep track of the log loss for both the training and validation sets so that we can plot these as a function of the epoch.

In your report provide:

1. Your plots of training and validation log loss vs epoch.
2. Assigning an observation to class 1 if the model outputs a value greater than 0.5, report the training and validation accuracy.

GRAPH:



Number of epochs: 10000
Training Accuracy: 84.63
Validation Accuracy: 89.73