Assignment1 (Fall 2024)

Total marks: 24

Instructions: Put all the answers in a single notebook

1. You have a training dataset with 60,000 images. How many samples per iteration (of gradient descent) you can have if you employ "mini-batch" gradient descent?
   a) 60,000 samples
   b) 20,000 samples
   c) 1 sample
   d) 1 < samples < 60,000

   1 mark

2. In your notebook, create a function which calculates the *sigmoid* function on an element or on an array. Pass a (4,1) array of [3, -25, -0.7, 25] as your input and calculate the output. Clearly print the output.

   1+1 marks

In the next set of questions, you will build a neural network with a single neuron for logistic regression on images from the Cat-images dataset. Use the ipynb file (Logistic_Regression_multilayer_network) as the template.

3. First use the load_data function to import the dataset. Use appropriate path for the import.
   (a) Write a simple set of instructions to show any image from the train_set_x subset. Print the corresponding label to check if the label is correctly showing a cat or not.

   2 marks

   (b) Next, find the values for number of training examples, number of test examples and the height/width of an image in the training set. Print each number.

   3 marks

4. Next, flatten the images in the training and test datasets into appropriate shapes.

   1 mark

5. Create a function for random initialization of weights. The function should assign fractional ($0<w<1$) random weights for a specified number of synapses. Initialize with zero bias. Return the weight values and the bias.

   2 marks

6. Define a function to calculate the forward and the backwards pass for the network. It should calculate the output of the neuron using sigmoid activation. Calculate the logistic loss, and the changes in weights/bias (dw, db).

   4 marks

7. Define another function to update the network parameters. Pass x, y, network parameters, number of iterations and learning rate as inputs to the function. Calculate the new weight and bias values in each iteration. Return the updated parameters, gradients and the cost.

   4 marks

8. Create the next function to convert the activations of the neuron into predictions (y). Use the following rule:

   if activation <= 0.5; prediction = 0
   else prediction = 1
   Return the predictions.

   2 marks

9. Finally, compile the model of the network. Use x_train, y_train, x_test, y_test, iterations, and learning rate as inputs to the function. Print the train and test accuracies return the cost and the predictions (for train as well as test data).

   3 marks