DATA ANALYTICS
GLASGOW

Large-Scale Computing for Data Analytics

Assignment 1

This assignment sheet is assessed and contributes 20% to your overall grade for Large-Scale Computing for Data Analytics. You can obtain a total of 20 marks for this assignment. You have to answer all questions on this sheet.

Please upload your answers before Sunday, June 20th, 11pm (UK time).

To upload your answers, log on to Moodle and go to Large-Scale Computing for Data Analytics. *Under* Coursework and Quizzes, *there is a link* Upload answers for assignment 1 (due June 20th). *Click on this link to upload the file containing your commented Python notebook.*

You should submit a Jupyter notebook (file extension `.ipynb`) which can be downloaded from a Google Colab by clicking File → Download .ipynb. *Please do not upload your code in other formats.*

Overview

For this assignment you will be analysing a dataset on human activity classification. The dataset consists of time series recordings from the inertial sensors of smartphones that are carried by people performing different activities.

You can read more about the dataset here and watch a video of the data collection here.

The dataset can be downloaded by using the following commands within a colab notebook:

```
# specify the data file name and url
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/00240/'
datafile = url + 'UCI%20HAR%20Dataset.zip'

# download the zip file from the web server using curl
!curl $datafile --output UCI_HAR_Dataset.zip

# unzip the file
!unzip -qq UCI_HAR_Dataset.zip

# change the directory name to remove spaces
!mv -f UCI\ HAR\ Dataset UCI_HAR_DATASET
```

The dataset contains both raw data and extracted features that can be used to classify human activities.

In this assignment you will be using multiclass logistic regression to train a classifier that will predict activity based on the features.

Task 1. Load the data using the command above. Read the text files contained in the folder you've downloaded that describe the dataset. Next write a couple of paragraphs in your notebook (in a Text cell) that summarise the data.

[1 marks]

Task 2. By looking at the labels in `test/y_test.txt` plot the time series data contained in the following files that can be found in the `Inertial Signals` folder:

```
    body_acc_x_test.txt
    body_acc_y_test.txt
    body_acc_z_test.txt
```

Plot one example of walking, sitting, standing, and laying for a randomly chosen data sample. You should produce 4 plots, each showing a random example of the behaviour with 3 lines representing the (x,y,z)-coordinates. Make sure to include a legend. [3 marks]

Task 3. Write Tensorflow code for performing multiclass logistic regression. Your code should include a prediction function that takes as inputs a tensor containing the 561 features associated with an array of samples and returns a predicted probability that each of the samples belongs to one of the 6 activity classes. Your code should also include a loss function that calculates the categorical cross entropy loss based on a set of predictions and a tensor containing one-hot encoded labels.

Hints: You can use standard tf.keras utilities to convert the data labels to one-hot encoding but note that the labels in the dataset are $[1, 2, 3, 4, 5, 6]$. Subtract 1 from the label to use standard 0-based indexing.

When calculating the loss, clip any predicted probabilities by using `tf.clip_by_value` so that all values are > 0.

[3 marks]

Task 4. Train the model using Stochastic Gradient Descent (`tf.keras.optimizers.SGD`) and the `X_train.txt` and `y_train.txt` data.

Use a learning rate of `1e-3` and run for 10000 iterations (initialise model weights and offsets to 0).

Plot the the loss over time and calculate the prediction accuracy on the training and test datasets by looking at the maximum probability value found with `np.argmax`.

[3 marks]

Task 5. Repeat the optimization task above except this time use the Adam optimizer. As before, calculate the train and test accuracy, and plot the loss over time. When optimizing the model make sure to start with weights and offsets set to 0, and not the values found using SGD.

[2 marks]

Task 6. Explain the differences you observe between the two different optimizers. (Type your answer directly into a Text cell in the notebook.)

[1 marks]

Task 7. Next, find the 50 features for each class that have the largest **absolute value** weights. These will be the features that have the largest contribution to the classification score for each activity.

Create a new muliclass logistic regression model using only the features that are in the top 50 list for at least one activity.

Here are some tensorflow functions that will be useful for this task:

```
tf.gather                     # can be used to extract slices of a tensor
                              # based on a list of indexes

tf.linalg.matrix_transpose    # transposes a tensor

tf.math.abs                   # returns the absolute values of a tensor

tf.math.top_k                 # finds the k largest values and their indices
                              # in the last dimension of a tensor

tf.reshape(X,[-1])            # reshapes a tensor, reshaping a tensor to a
                              # shape of [-1] will flatten the tensor

tf.unique                     # returns the unique elements of a tensor
```

Note, that while you will be selecting the top 50 features for each activity you should **not** end up with 300 features. Many of the features will be important for multiple activities so you will need to remove any duplicate entries once you've found the top 50 for each activity.

[4 marks]

Task 8. Optimize the model using the features you've found in the previous task. Use the Adam optimizer with the same parameters as before and print out the test-train accuracy and plot the loss function. Within a Text cell, comment on the results you obtain.

[3 marks]

Total : 20