ML0120EN-1.2-Review-LinearRegressionwithTensorFlow

October 17, 2020

LINEAR REGRESSION WITH TENSORFLOW

LINEAR REGRESSION WITH TENSORFLOW

Objective for this Notebook

1. What is Linear Regression
   2. Linear Regression with TensorFlow.

In this notebook we will overview the implementation of Linear Regression with TensorFlow

Table of Contents

Linear Regression

Defining a linear regression in simple terms, is the approximation of a linear model used to describe the relationship between two or more variables. In a simple linear regression there are two variables, the dependent variable, which can be seen as the “state” or “final goal” that we study and try to predict, and the independent variables, also known as explanatory variables, which can be seen as the “causes” of the “states”.

When more than one independent variable is present the process is called multiple linear regression. When multiple dependent variables are predicted the process is known as multivariate linear regression.

The equation of a simple linear model is

$$Y = aX + b$$

Where Y is the dependent variable and X is the independent variable, and a and b being the parameters we adjust. a is known as “slope” or “gradient” and b is the “intercept”. You can interpret this equation as Y being a function of X, or Y being dependent on X.

If you plot the model, you will see it is a line, and by adjusting the “slope” parameter you will change the angle between the line and the independent variable axis, and the “intercept parameter” will affect where it crosses the dependent variable’s axis.

We begin by installing TensorFlow version 2.2.0 and its required prerequistes.

[1]:

```
!pip install grpcio==1.24.3
!pip install tensorflow==2.2.0
```

```
Collecting grpcio==1.24.3
  Downloading https://files.pythonhosted.org/packages/30/54/c9810421e41ec0
bca2228c6f06b1b1189b196b69533cbcac9f71b44727f8/grpcio-1.24.3-cp36-cp36m-manylinu
x2010_x86_64.whl (2.2MB)
   |                 | 2.2MB 9.7MB/s eta 0:00:01
Requirement already satisfied: six>=1.5.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
grpcio==1.24.3) (1.15.0)
Installing collected packages: grpcio
  Found existing installation: grpcio 1.31.0
    Uninstalling grpcio-1.31.0:
      Successfully uninstalled grpcio-1.31.0
Successfully installed grpcio-1.24.3
Collecting tensorflow==2.2.0
  Downloading https://files.pythonhosted.org/packages/3d/be/679ce5254a8c8d
07470efb4a4c00345fae91f766e64f1c2aece8796d7218/tensorflow-2.2.0-cp36-cp36m-manyl
inux2010_x86_64.whl (516.2MB)
   |                 | 516.2MB 37kB/s  eta 0:00:014  |
| 962kB 6.7MB/s eta 0:01:18     |                                | 2.6MB 6.7MB/s
eta 0:01:18                              | 9.7MB 6.7MB/s eta 0:01:17     |
| 21.0MB 34.5MB/s eta 0:00:15     |                              | 75.9MB
5.4MB/s eta 0:01:22MB/s eta 0:01:00     |                            |
170.4MB 9.6MB/s eta 0:00:36     |                            | 171.2MB
9.6MB/s eta 0:00:36     |                            | 172.8MB 9.6MB/s eta
0:00:36     |                            | 175.1MB 9.6MB/s eta 0:00:36
|                           | 177.3MB 9.6MB/s eta 0:00:36     |
| 178.7MB 7.4MB/s eta 0:00:46     |                            | 181.8MB
7.4MB/s eta 0:00:46     |                           | 184.0MB 7.4MB/s eta
0:00:46     |                           | 190.2MB 7.4MB/s eta 0:00:45
|                           | 193.3MB 6.6MB/s eta 0:00:49     |
| 194.1MB 6.6MB/s eta 0:00:49     |                            | 201.4MB
6.6MB/s eta 0:00:48     |                           | 202.1MB 6.6MB/s eta
0:00:48     |                          | 212.1MB 5.9MB/s eta 0:00:52
|                         | 236.0MB 39.1MB/s eta 0:00:08
|                        | 261.0MB 39.5MB/s eta 0:00:07
| 277.8MB 42.8MB/s eta 0:00:06     |                         | 283.7MB
43.3MB/s eta 0:00:06                    | 300.2MB 39.3MB/s eta
0:00:06                      | 302.2MB 39.3MB/s eta
0:00:06                      | 305.2MB 39.3MB/s eta
0:00:06                      | 307.5MB 39.3MB/s eta 0:00:06MB/s eta
0:00:05                 | 329.0MB 45.0MB/s eta 0:00:05             | 331.5MB
45.0MB/s eta 0:00:05     |                      | 332.3MB 45.0MB/s eta
0:00:05                 | 333.9MB 45.0MB/s eta 0:00:05
|                      | 351.5MB 43.5MB/s eta 0:00:04:00:047MB 5.4MB/s
```

```
eta 0:00:24            | 394.5MB 5.4MB/s eta 0:00:23
|                     | 411.2MB 5.5MB/s eta 0:00:20    | 442.1MB
5.5MB/s eta 0:00:14           | 457.3MB 5.5MB/s eta 0:00:11           |
469.5MB 9.6MB/s eta 0:00:05                | 470.1MB 9.6MB/s eta
0:00:05      |                  | 495.7MB 5.5MB/s eta 0:00:04
|                  | 504.9MB 5.4MB/s eta 0:00:033.6MB 5.5MB/s eta
0:00:01
Collecting google-pasta>=0.1.8 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/a3/de/c648ef6835192e
6e2cc03f40b19eeda4382c49b5bafb43d88b931c4c74ac/google_pasta-0.2.0-py3-none-
any.whl (57kB)
     |                  | 61kB 21.7MB/s eta 0:00:01
Requirement already satisfied: protobuf>=3.8.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorflow==2.2.0) (3.13.0)
Collecting tensorflow-estimator<2.3.0,>=2.2.0 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/a4/f5/926ae53d6a226e
c0fda5208e0e581cffed895ccc89e36ba76a8e60895b78/tensorflow_estimator-2.2.0-py2.py
3-none-any.whl (454kB)
     |                  | 460kB 38.6MB/s eta 0:00:01
Requirement already satisfied: six>=1.12.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorflow==2.2.0) (1.15.0)
Collecting scipy==1.4.1; python_version >= "3" (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/dc/29/162476fd442031
16e7980cfbd9352eef9db37c49445d1fec35509022f6aa/scipy-1.4.1-cp36-cp36m-manylinux1
_x86_64.whl (26.1MB)
     |                  | 26.1MB 5.5MB/s eta 0:00:01    |
| 4.9MB 8.9MB/s eta 0:00:03MB/s eta 0:00:02
Requirement already satisfied: absl-py>=0.7.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorflow==2.2.0) (0.10.0)
Requirement already satisfied: termcolor>=1.1.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorflow==2.2.0) (1.1.0)
Collecting gast==0.3.3 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/d6/84/759f5dd23fec8ba71952
d97bcc7e2c9d7d63bdc582421f3cd4be845f0c98/gast-0.3.3-py2.py3-none-any.whl
Requirement already satisfied: grpcio>=1.8.6 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorflow==2.2.0) (1.24.3)
Collecting h5py<2.11.0,>=2.10.0 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/60/06/cafdd44889200e
5438b897388f3075b52a8ef01f28a17366d91de0fa2d05/h5py-2.10.0-cp36-cp36m-manylinux1
_x86_64.whl (2.9MB)
     |                  | 2.9MB 14.5MB/s eta 0:00:01
Collecting tensorboard<2.3.0,>=2.2.0 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/1d/74/0a6fcb206dcc72
```

```
a6da9a62dd81784bfdbff5fedb099982861dc2219014fb/tensorboard-2.2.2-py3-none-
any.whl (3.0MB)
     |                 | 3.0MB 36.3MB/s eta 0:00:01
Collecting keras-preprocessing>=1.1.0 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/79/4c/7c3275a01e12ef
9368a892926ab932b33bb13d55794881e3573482b378a7/Keras_Preprocessing-1.1.2-py2.py3
-none-any.whl (42kB)
     |                 | 51kB 24.2MB/s eta 0:00:01
Requirement already satisfied: numpy<2.0,>=1.16.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorflow==2.2.0) (1.19.1)
Collecting wrapt>=1.11.1 (from tensorflow==2.2.0)
  Using cached https://files.pythonhosted.org/packages/82/f7/e43cefbe88c5fd371f4
cf0cf5eb3feccd07515af9fd6cf7dbf1d1793a797/wrapt-1.12.1.tar.gz
Collecting opt-einsum>=2.3.2 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/bc/19/404708a7e54ad2
798907210462fd950c3442ea51acc8790f3da48d2bee8b/opt_einsum-3.3.0-py3-none-any.whl
(65kB)
     |                 | 71kB 4.8MB/s  eta 0:00:01
Requirement already satisfied: wheel>=0.26; python_version >= "3" in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorflow==2.2.0) (0.35.1)
Collecting astunparse==1.6.3 (from tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/2b/03/13dde6512ad7b4557eb7
92fbcf0c653af6076b81e5941d36ec61f7ce6028/astunparse-1.6.3-py2.py3-none-any.whl
Requirement already satisfied: setuptools in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
protobuf>=3.8.0->tensorflow==2.2.0) (49.6.0.post20200917)
Requirement already satisfied: werkzeug>=0.11.15 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (1.0.1)
Requirement already satisfied: requests<3,>=2.21.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (2.24.0)
Collecting google-auth-oauthlib<0.5,>=0.4.1 (from
tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/7b/b8/88def36e74bee9fce511
c9519571f4e485e890093ab7442284f4ffaef60b/google_auth_oauthlib-0.4.1-py2.py3-none
-any.whl
Collecting google-auth<2,>=1.6.3 (from
tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/1f/cf/724b6436967a8b
e879c8de16b09fd80e0e7b0bcad462f5c09ee021605785/google_auth-1.22.1-py2.py3-none-
any.whl (114kB)
     |                 | 122kB 44.6MB/s eta 0:00:01
Collecting tensorboard-plugin-wit>=1.6.0 (from
tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/b6/85/5c5ac0a8c5efdf
```

```
ab916e9c6bc18963f6a6996a8a1e19ec4ad8c9ac9c623c/tensorboard_plugin_wit-1.7.0-py3-
none-any.whl (779kB)
     |                 | 788kB 38.5MB/s eta 0:00:01
Requirement already satisfied: markdown>=2.6.8 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (3.2.2)
Requirement already satisfied: idna<3,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (2020.6.20)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (1.25.10)
Requirement already satisfied: chardet<4,>=3.0.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
requests<3,>=2.21.0->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (3.0.4)
Collecting requests-oauthlib>=0.7.0 (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0)
  Using cached https://files.pythonhosted.org/packages/a3/12/b92740d845ab62ea4ed
f04d2f4164d82532b5a0b03836d4d4e71c6f3d379/requests_oauthlib-1.3.0-py2.py3-none-
any.whl
Collecting cachetools<5.0,>=2.0.0 (from google-
auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0)
  Using cached https://files.pythonhosted.org/packages/cd/5c/f3aa86b6d5482f3051b
433c7616668a9b96fbe49a622210e2c9781938a5c/cachetools-4.1.1-py3-none-any.whl
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from google-
auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (0.2.8)
Collecting rsa<5,>=3.1.4; python_version >= "3.5" (from google-
auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0)
  Using cached https://files.pythonhosted.org/packages/1c/df/c3587a667d6b308fadc
90b99e8bc8774788d033efcc70f4ecaae7fad144b/rsa-4.6-py3-none-any.whl
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
markdown>=2.6.8->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (2.0.0)
Collecting oauthlib>=3.0.0 (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0)
  Downloading https://files.pythonhosted.org/packages/05/57/ce2e7a8fa7c0af
b54a0581b14a65b56e62b5759dbc98e80627142b8a3704/oauthlib-3.1.0-py2.py3-none-
any.whl (147kB)
     |                 | 153kB 6.8MB/s eta 0:00:01
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
pyasn1-modules>=0.2.1->google-
auth<2,>=1.6.3->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (0.4.8)
Requirement already satisfied: zipp>=0.5 in
```

```
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from importlib-
metadata; python_version <
"3.8"->markdown>=2.6.8->tensorboard<2.3.0,>=2.2.0->tensorflow==2.2.0) (3.2.0)
Building wheels for collected packages: wrapt
  Building wheel for wrapt (setup.py) … done
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/b1/c2/ed/d622082
60edbd3fa7156545c00ef966f45f2063d0a84f8208a
Successfully built wrapt
Installing collected packages: google-pasta, tensorflow-estimator, scipy, gast,
h5py, oauthlib, requests-oauthlib, cachetools, rsa, google-auth, google-auth-
oauthlib, tensorboard-plugin-wit, tensorboard, keras-preprocessing, wrapt, opt-
einsum, astunparse, tensorflow
  Found existing installation: scipy 1.5.2
    Uninstalling scipy-1.5.2:
      Successfully uninstalled scipy-1.5.2
  Found existing installation: gast 0.4.0
    Uninstalling gast-0.4.0:
      Successfully uninstalled gast-0.4.0
  Found existing installation: h5py 2.8.0
    Uninstalling h5py-2.8.0:
      Successfully uninstalled h5py-2.8.0
  Found existing installation: tensorboard 1.8.0
    Uninstalling tensorboard-1.8.0:
      Successfully uninstalled tensorboard-1.8.0
  Found existing installation: tensorflow 1.8.0
    Uninstalling tensorflow-1.8.0:
      Successfully uninstalled tensorflow-1.8.0
Successfully installed astunparse-1.6.3 cachetools-4.1.1 gast-0.3.3 google-
auth-1.22.1 google-auth-oauthlib-0.4.1 google-pasta-0.2.0 h5py-2.10.0 keras-
preprocessing-1.1.2 oauthlib-3.1.0 opt-einsum-3.3.0 requests-oauthlib-1.3.0
rsa-4.6 scipy-1.4.1 tensorboard-2.2.2 tensorboard-plugin-wit-1.7.0
tensorflow-2.2.0 tensorflow-estimator-2.2.0 wrapt-1.12.1
```

Restart kernel for latest version of TensorFlow to be activated

Next, let's first import the required packages:

[2]:

```python
import tensorflow as tf
import numpy as np
import pandas as pd
import pylab as pl
import matplotlib.patches as mpatches
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams['figure.figsize'] = (10, 6)
```

[3]:

```python
if not tf.__version__ == '2.2.0':
    print(tf.__version__)
    raise ValueError('please upgrade to TensorFlow 2.2.0, or restart your␣
↪Kernel (Kernel->Restart & Clear Output)')
```

IMPORTANT! => Please restart the kernel by clicking on “Kernel”->“Restart and Clear Outout” and wait until all output disapears. Then your changes are beeing picked up

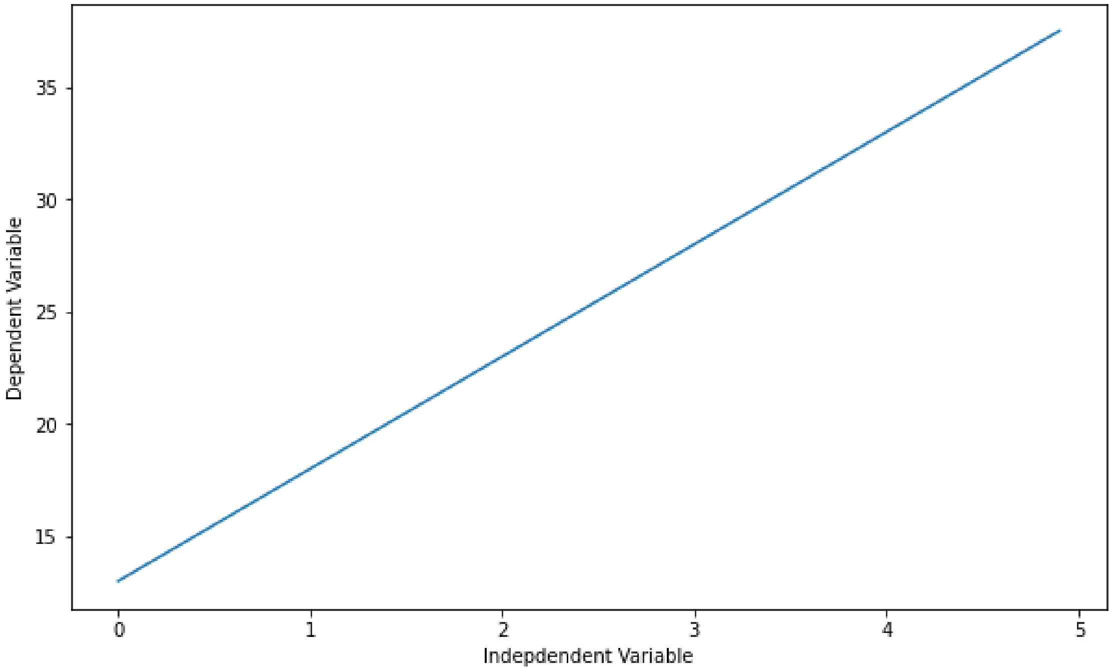
Let’s define the independent variable:

[4]:
```
X = np.arange(0.0, 5.0, 0.1)
X
```

[4]:
```
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
       1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1, 2.2, 2.3, 2.4, 2.5,
       2.6, 2.7, 2.8, 2.9, 3. , 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8,
       3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9])
```

[6]:
```
a = 5
b = 13
Y = a * X + b

plt.plot(X, Y)
plt.ylabel('Dependent Variable')
plt.xlabel('Indepdendent Variable')
plt.show()
```



OK… but how can we see this concept of linear relations with a more meaningful point of view?

Simple linear relations were used to try to describe and quantify many observable physical phenomena, the easiest to understand are speed and distance traveled:

$$DistanceTraveled = SpeedtimesTime + InitialDistance$$

$$Speed = AccelerationtimesTime + InitialSpeed$$

They are also used to describe properties of different materials:

$$Force = DeformationtimesStiffness$$

$$HeatTransfered = TemperatureDifferencetimesThermalConductivity$$

$$ElectricalTension(Voltage) = ElectricalCurrenttimesResistance$$

$$Mass = VolumetimesDensity$$

When we perform an experiment and gather the data, or if we already have a dataset and we want to perform a linear regression, what we will do is adjust a simple linear model to the dataset, we adjust the "slope" and "intercept" parameters to the data the best way possible, because the closer the model comes to describing each ocurrence, the better it will be at representing them.

So how is this "regression" performed?

Linear Regression with TensorFlow

A simple example of a linear function can help us understand the basic mechanism behind TensorFlow.

For the first part we will use a sample dataset, and then we'll use TensorFlow to adjust and get the right parameters. We download a dataset that is related to fuel consumption and Carbon dioxide emission of cars.

[7]:
```
!wget -O FuelConsumption.csv https://s3-api.us-geo.objectstorage.softlayer.net/
 ↪cf-courses-data/CognitiveClass/ML0101ENv3/labs/FuelConsumptionCo2.csv
```

```
--2020-10-17 01:09:26--  https://s3-api.us-geo.objectstorage.softlayer.net/cf-
courses-data/CognitiveClass/ML0101ENv3/labs/FuelConsumptionCo2.csv
Resolving s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-
geo.objectstorage.softlayer.net)… 67.228.254.196
Connecting to s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-
geo.objectstorage.softlayer.net)|67.228.254.196|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 72629 (71K) [text/csv]
Saving to: 'FuelConsumption.csv'
```

```
FuelConsumption.csv 100%[===================>]  70.93K  --.-KB/s    in 0.06s

2020-10-17 01:09:26 (1.26 MB/s) - ‘FuelConsumption.csv’ saved [72629/72629]
```

Understanding the Data

FuelConsumption.csv:

We have downloaded a fuel consumption dataset, FuelConsumption.csv, which contains model-specific fuel consumption ratings and estimated carbon dioxide emissions for new light-duty vehicles for retail sale in Canada. Dataset source

- **MODELYEAR** e.g. 2014
- **MAKE** e.g. Acura
- **MODEL** e.g. ILX
- **VEHICLE CLASS** e.g. SUV
- **ENGINE SIZE** e.g. 4.7
- **CYLINDERS** e.g 6
- **TRANSMISSION** e.g. A6
- **FUEL CONSUMPTION in CITY(L/100 km)** e.g. 9.9
- **FUEL CONSUMPTION in HWY (L/100 km)** e.g. 8.9
- **FUEL CONSUMPTION COMB (L/100 km)** e.g. 9.2
- **CO2 EMISSIONS (g/km)** e.g. 182 –> low –> 0

[8]:
```
df = pd.read_csv("FuelConsumption.csv")
df.head()
```

[8]:
```
   MODELYEAR   MAKE       MODEL VEHICLECLASS  ENGINESIZE  CYLINDERS   \
0       2014  ACURA         ILX      COMPACT         2.0          4
1       2014  ACURA         ILX      COMPACT         2.4          4
2       2014  ACURA  ILX HYBRID      COMPACT         1.5          4
3       2014  ACURA     MDX 4WD  SUV - SMALL         3.5          6
4       2014  ACURA     RDX AWD  SUV - SMALL         3.5          6

  TRANSMISSION FUELTYPE  FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY   \
0          AS5        Z                   9.9                  6.7
1           M6        Z                  11.2                  7.7
2          AV7        Z                   6.0                  5.8
3          AS6        Z                  12.7                  9.1
4          AS6        Z                  12.1                  8.7

   FUELCONSUMPTION_COMB  FUELCONSUMPTION_COMB_MPG  CO2EMISSIONS
0                   8.5                        33           196
1                   9.6                        29           221
2                   5.9                        48           136
3                  11.1                        25           255
4                  10.6                        27           244
```

Lets say we want to use linear regression to predict Co2Emission of cars based on their engine size. So, lets define X and Y value for the linear regression, that is, train_x and train_y:

[9]:
```
train_x = np.asanyarray(df[['ENGINESIZE']])
train_y = np.asanyarray(df[['CO2EMISSIONS']])
```

First, we initialize the variables a and b, with any random guess, and then we define the linear function:

[18]:
```
a = tf.Variable(np.random.rand())
b = tf.Variable(np.random.rand())

def h(x):
   y = a*x + b
   return y
```

Now, we are going to define a loss function for our regression, so we can train our model to better fit our data. In a linear regression, we minimize the squared error of the difference between the predicted values(obtained from the equation) and the target values (the data that we have). In other words we want to minimize the square of the predicted values minus the target value. So we define the equation to be minimized as loss.

To find value of our loss, we use tf.reduce_mean(). This function finds the mean of a multidimensional tensor, and the result can have a different dimension.

[20]:
```
def loss_object(y,train_y) :
    return tf.reduce_mean(tf.square(y - train_y))

loss_object_TF = tf.keras.losses.MeanSquaredLogarithmicError() # predefined␣
 ↪method offered by TensorFlow to calculate loss function
```

Now we are ready to start training and run the graph. We use GradientTape to calculate gradients:

[21]:
```
learning_rate = 0.01
train_data = []
loss_values =[]
training_epochs = 200

for epoch in range(training_epochs):
    with tf.GradientTape() as tape:
        y_predicted = h(train_x)
        loss_value = loss_object(train_y,y_predicted)
        loss_values.append(loss_value)
        gradients = tape.gradient(loss_value, [b,a])
        b.assign_sub(gradients[0]*learning_rate)
        a.assign_sub(gradients[1]*learning_rate)
        if epoch % 5 == 0:
            train_data.append([a, b])
```

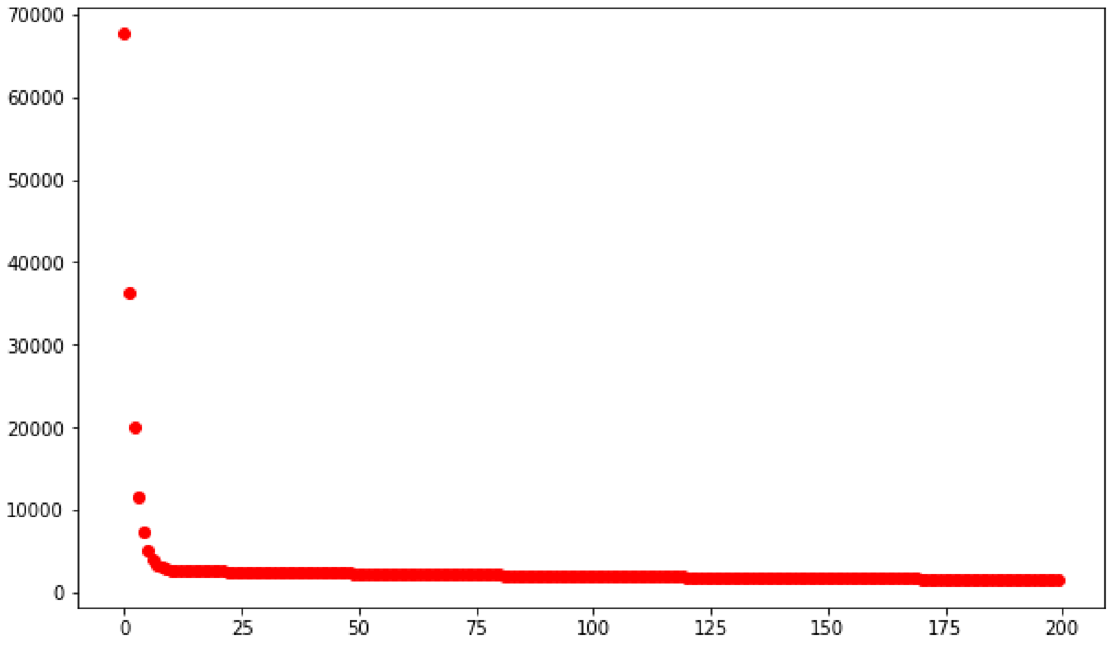
Lets plot the loss values to see how it has changed during the training:

[22]:

```
plt.plot(loss_values, 'ro')
```

[22]:

```
[<matplotlib.lines.Line2D at 0x7f96c48f43c8>]
```



Lets visualize how the coefficient and intercept of line has changed to fit the data:

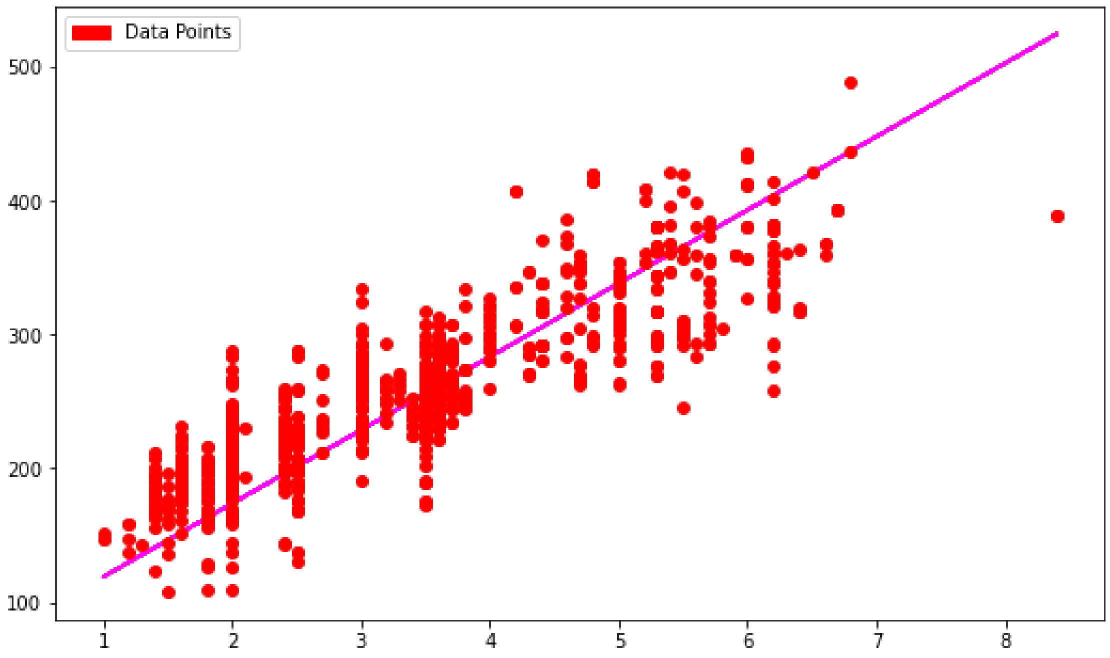
[23]:

```
cr, cg, cb = (1.0, 1.0, 0.0)
for f in train_data:
    cb += 1.0 / len(train_data)
    cg -= 1.0 / len(train_data)
    if cb > 1.0: cb = 1.0
    if cg < 0.0: cg = 0.0
    [a, b] = f
    f_y = np.vectorize(lambda x: a*x + b)(train_x)
    line = plt.plot(train_x, f_y)
    plt.setp(line, color=(cr,cg,cb))

plt.plot(train_x, train_y, 'ro')
green_line = mpatches.Patch(color='red', label='Data Points')

plt.legend(handles=[green_line])

plt.show()
```



0.1 Want to learn more?

Running deep learning programs usually needs a high performance platform. **PowerAI** speeds up deep learning and AI. Built on IBM's Power Systems, **PowerAI** is a scalable software platform that accelerates deep learning and AI with blazing performance for individual users or enterprises. The **PowerAI** platform supports popular machine learning libraries and dependencies including TensorFlow, Caffe, Torch, and Theano. You can use PowerAI on IMB Cloud.

Also, you can use **Watson Studio** to run these notebooks faster with bigger datasets.**Watson Studio** is IBM's leading cloud solution for data scientists, built by data scientists. With Jupyter notebooks, RStudio, Apache Spark and popular libraries pre-packaged in the cloud, **Watson Studio** enables data scientists to collaborate on their projects without having to install anything. Join the fast-growing community of **Watson Studio** users today with a free account at Watson Studio.This is the end of this lesson. Thank you for reading this notebook, and good luck on your studies.

0.1.1 Thanks for completing this lesson!

If you are familiar with some of these methods and concepts, this tutorial might have been boring for you, but it is important to get used to the TensorFlow mechanics, and feel familiar and comfortable using it, so you can build more complex algorithms in it.

Created by Romeo Kienzler , Saeed Aghabozorgi , Rafael Belo Da Silva

Updated to TF 2.X by Samaya Madhavan

0.2 Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-09-21	2.0	Srishti	Migrated Lab to Markdown and added to course repo in GitLab

##