Task-02

August 4, 2022

0.1 Task: 02

The goal of this task is to implement **Gradient Descent** algorithm in Python. Gradient Descent dictates how the weights get updated from an inital value to ensure we reach a minimal loss value.

[2]:
```
#import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Let us make use of a randomly-created sample dataset as follows

[3]:
```
#sample-dataset
x_data = [1.0, 2.0, 3.0]
y_data = [2.0, 4.0, 6.0]

# Let us initialize our weight value (w) with 1.0
w = 1.0
```

0.2 Task: 02 - a

Implement the forward and loss functions

[4]:
```
#forward function to calculate y_pred for a given x according to the linear␣
 ↪model defined above
def forward(x):
    #implement the forward model to compute y_pred as w*x
    ## YOUR CODE STARTS HERE
    return (w*x)

    ## YOUR CODE ENDS HERE

#loss-function to compute the mean-squared error between y_pred and y_actual
def loss(y_pred, y_actual):
    #calculate the mean-squared-error between y_pred and y_actual
    ## YOUR CODE STARTS HERE
    return ((y_pred - y_actual)**2)

    ## YOUR CODE ENDS HERE
```

0.2.1 Gradient Descent

We update the `w` such that loss is minimum. The factor by which `w` is updated each time is called `alpha(learning rate)`.

New `w` is `w` minus `alpha` times derivative of `loss` against `w`, which can be mathematically expressed as follows:

$w = w - \alpha * \frac{d(loss)}{dw}$

This equation is dependent on how the loss function has been defined. In the current case below formula will dictate how to update the value of w for each pass.

$w = w - \alpha * (2x) * (y_{pred} - y_{actual})$

0.3 Task: 02 - b

Complete the gradient function

[8]:
```
# Function to calculate the gradient for w to be updated and get min loss.
# Gradient = derivative of the loss for constant x and y with respect to the␣
 ↪weight w

def gradient(x,y):
    #implement the gradient of loss with respect to the weight (w)
    ## YOUR CODE STARTS HERE
    return (2*x)*(x-y)

    ## YOUR CODE ENDS HERE
```

Calculate y_{pred} for $x = 4$ without training the model

[5]:
```
y_pred_without_train = forward(4)
```

Begin Training

[10]:
```
# In this method, we learn the dataset multiple times (called epochs)
# Each time, the weight (w) gets updates using the graident decent algorithm␣
 ↪based on weights of the previous epoch

alpha = 0.01 # Let us set learning rate as 0.01
weight_list = []
loss_list=[]

for epoch in range(100):
    total_loss=0
    count = 0
    for x, y in zip(x_data, y_data):

        #call the forward function to calculate y_pred
        ## YOUR CODE STARTS HERE
```

```
        y_pred = forward(x)
        ## YOUR CODE ENDS HERE

        #call the gradient function to obtain the grad for the given data-pair␣
↪and update the weight
        ## YOUR CODE STARTS HERE
        grad = gradient(y_pred,y)
        w = w - ((alpha) * grad)
        ## YOUR CODE ENDS HERE

        #call the loss function to obtain the mean_squared_error of the current␣
↪sample
        ## YOUR CODE STARTS HERE
        current_loss = loss(y_pred , y)
        ## YOUR CODE ENDS HERE

        total_loss+=current_loss

        count += 1

    avg_mse = total_loss / count
    print('Progress: ', epoch, 'w=', w, 'loss=', avg_mse)
    weight_list.append(w)
    loss_list.append(avg_mse)
```

```
Progress:  0 w= 1.2781691518156801 loss= 4.044039469738666
Progress:  1 w= 1.5249963751247337 loss= 2.025612224472239
Progress:  2 w= 1.7122186064854195 loss= 0.8471382579194353
Progress:  3 w= 1.8362951427474148 loss= 0.30287559461054236
Progress:  4 w= 1.9106973615072778 loss= 0.0963204167479248
Progress:  5 w= 1.9524954696789971 loss= 0.028366810978317003
Progress:  6 w= 1.97508516851224 loss= 0.007980313256689919
Progress:  7 w= 1.9870324934460457 loss= 0.0021882573516342164
Progress:  8 w= 1.9932780346715335 loss= 0.0005917967107410784
Progress:  9 w= 1.9965229051743878 loss= 0.00015888186040234655
Progress:  10 w= 1.9982033684372251 loss= 4.249311599791234e-05
Progress:  11 w= 1.9990722006304151 loss= 1.13423206331625e-05
Progress:  12 w= 1.9995210158018568 loss= 3.024401664462905e-06
Progress:  13 w= 1.999752758021285 loss= 8.060213583108322e-07
Progress:  14 w= 1.9998723887065253 loss= 2.1475069770097435e-07
Progress:  15 w= 1.999934137472199 loss= 5.7208577411083136e-08
Progress:  16 w= 1.9999660078537662 loss= 1.5238982918796542e-08
Progress:  17 w= 1.9999824565871027 loss= 4.059143807590572e-09
Progress:  18 w= 1.9999909458609209 loss= 1.0811960242165275e-09
Progress:  19 w= 1.9999953271806927 loss= 2.8798514041568e-10
Progress:  20 w= 1.9999975883728258 loss= 7.670672962343803e-11
Progress:  21 w= 1.9999987553678442 loss= 2.0431285508883732e-11
```

```
Progress:  22 w= 1.9999993576500519 loss= 5.441984725845563e-12
Progress:  23 w= 1.9999996684856889 loss= 1.4495013855707174e-12
Progress:  24 w= 1.9999998289067709 loss= 3.860822111635735e-13
Progress:  25 w= 1.9999999116994638 loss= 1.0283497958077339e-13
Progress:  26 w= 1.999999954428445 loss= 2.739062199221258e-14
Progress:  27 w= 1.9999999764807024 loss= 7.295631774062847e-15
Progress:  28 w= 1.9999999878617845 loss= 1.943228676846626e-15
Progress:  29 w= 1.9999999937355155 loss= 5.175888146205401e-16
Progress:  30 w= 1.9999999967669246 loss= 1.378624248252959e-16
Progress:  31 w= 1.9999999983314227 loss= 3.67203570256264e-17
Progress:  32 w= 1.999999999138854 loss= 9.780654519758389e-18
Progress:  33 w= 1.999999999555566 loss= 2.605126923387936e-18
Progress:  34 w= 1.9999999997706295 loss= 6.938886226625894e-19
Progress:  35 w= 1.999999999881623 loss= 1.8482052645257509e-19
Progress:  36 w= 1.999999999938906 loss= 4.922766865250748e-20
Progress:  37 w= 1.9999999999684697 loss= 1.3112058091910297e-20
Progress:  38 w= 1.9999999999837272 loss= 3.492485369288335e-21
Progress:  39 w= 1.9999999999916018 loss= 9.30264444546223e-22
Progress:  40 w= 1.9999999999956657 loss= 2.477701598348826 4e-22
Progress:  41 w= 1.9999999999977631 loss= 6.599704108919093e-23
Progress:  42 w= 1.9999999999988456 loss= 1.7578332569969548e-23
Progress:  43 w= 1.9999999999994043 loss= 4.680997090022276e-24
Progress:  44 w= 1.9999999999996925 loss= 1.2471834540858678e-24
Progress:  45 w= 1.9999999999998412 loss= 3.324675485690782e-25
Progress:  46 w= 1.999999999999918 loss= 8.841147091042956e-26
Progress:  47 w= 1.9999999999999578 loss= 2.355224762088072e-26
Progress:  48 w= 1.9999999999999782 loss= 6.248370015029329e-27
Progress:  49 w= 1.9999999999999887 loss= 1.6540441030221565e-27
Progress:  50 w= 1.9999999999999942 loss= 4.532827630604318e-28
Progress:  51 w= 1.9999999999999971 loss= 1.1951242714098329e-28
Progress:  52 w= 1.9999999999999987 loss= 2.9072811277832704e-29
Progress:  53 w= 1.9999999999999993 loss= 7.165486555757524e-30
Progress:  54 w= 1.9999999999999996 loss= 1.791371638939381e-30
Progress:  55 w= 1.9999999999999998 loss= 1.3805065841367707e-30
Progress:  56 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  57 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  58 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  59 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  60 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  61 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  62 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  63 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  64 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  65 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  66 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  67 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  68 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  69 w= 1.9999999999999998 loss= 3.4512664603419266e-31
```

```
Progress:  70 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  71 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  72 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  73 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  74 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  75 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  76 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  77 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  78 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  79 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  80 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  81 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  82 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  83 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  84 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  85 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  86 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  87 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  88 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  89 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  90 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  91 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  92 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  93 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  94 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  95 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  96 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  97 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  98 w= 1.9999999999999998 loss= 3.4512664603419266e-31
Progress:  99 w= 1.9999999999999998 loss= 3.4512664603419266e-31
```

Calculate y_{pred} for $x = 4$ after training the model

[11]:
```
y_pred_with_train = forward(4)

print("Actual Y Value for x=4 : 8")
print("Predicted Y Value before training : " , y_pred_without_train)
print("Predicted Y Value after training : " , y_pred_with_train)
```

```
Actual Y Value for x=4 : 8
Predicted Y Value before training :  4.0
Predicted Y Value after training :  7.999999999999999
```

0.3.1 Visualize Loss as a function of weight

[12]:
```
plt.plot(weight_list, loss_list)
plt.ylabel('Loss')
plt.xlabel('w')
plt.show()
```